CHASE

J.P.Morgan

Orbital Gateway Digital Wallet API Developer Guide

January 2024

Version 2.6.2.0

What's new in version 2.6.2.0

The following updates have been made to this document since version 2.6.1.0:

Effective upon publication

- Updated the Authorization time frame windows in the Mark for Capture section.
- Updated JCB information in the Cryptograms section.
- Updated the `digitalTokenCryptogram` field description in the Request elements – Mark for Capture section and added `digitalTokenCryptogram` field to the Request elements – Auth and Auth and Capture section.

Table of contents

About this guide

This guide provides the steps for implementing a digital wallet into a merchant's mobile app or website using the Merchant Services Digital Wallet Application Programming Interface (API), also referred to as the Digital Wallet API.

Note: Meta Checkout is currently in pilot. Consult your J.P. Morgan Relationship Manager prior to development.

References

Refer to the following documents for additional information regarding Consumer Digital Payment Tokens (CDPTs) and digital wallets:

- Stratus Online Processing Developer Guide
- Stratus 120-Byte Batch Processing Developer Guide
- Orbital Gateway XML Interface Developer Guide
- Orbital Gateway Web Service Interface Developer Guide
- Tandem PNS ISO Format Developer Guide
- Tandem UTF Host Capture Developer Guide
- Tandem TCS Batch File Developer Guide

Audience

Primary users of this document are the merchants' developers and technical support staff. Primary users must have a working knowledge of JavaScript Object Notation (JSON) data interchange formats.

Getting help

A J.P. Morgan Technical Implementations Analyst is available to assist merchants in a digital wallet implementation. Contact the J.P. Morgan Account Executive or Relationship Manager for additional information.

Terminology

Authorization and Capture Requests

Authorization and Capture requests allow merchants to confirm whether a customer has submitted a valid method of payment with their order, as well as request that the funds be deposited into a merchant's account. Merchants can use Authorization and Capture requests for direct sale transactions that do not require future fulfillments.

Authorization Requests

Authorization requests allow merchants to confirm whether a customer has submitted a valid method of payment with their order, as well as determine whether they have funds sufficient enough to issue the goods or services. Merchants can use an Authorization request if the items in the transaction will be billed at a later date, or at the time of actual fulfilment or shipment.

Consumer Digital Payment Token (CDPT)

Issuer consortiums and payment brands work together to develop and set industry standards for token transactions. A token is a surrogate value that resides in digital wallet applications and replaces cardholder account numbers during transaction processing.

Certificate Signing Request (CSR)

A CSR is a message sent from an applicant to a certificate authority to apply for a digital identity certificate. CSRs contain public keys, identifying information, and integrity protection, and is one of the first steps toward establishing a merchant SSL certificate.

Cryptogram

A cryptogram is a unique alphanumeric string that is included in the encrypted bundle of a digital wallet payload. A new cryptogram is generated with each payment request.

Debundle Only

A Debundle Only request allows merchants to send J.P. Morgan encrypted payloads for decryption. Decrypted bundles can be sent to one of J.P. Morgan's host systems for processing.

Digital Primary Account Number (DPAN)

The network token that is generated by the payment network or issuer who identifies the provisioned card that is associated with a cardholder's Funding Primary Account Number (FPAN).

Funding Primary Account Number (FPAN)

The cardholder account number that is available on the face of the physical card. Most digital wallet transactions are processed with a DPAN instead of the FPAN to provide additional security to the consumer.

Issuer

A financial institution that issues cards to consumers or businesses.

Mark for Capture (MFC)

An MFC marks a previously authorized transaction as being ready for clearing, and is present for future fulfillment models. A transaction can be authorized now and marked for capture at any time within a four-month span following the time of transaction.

Payload

The payment details that are securely passed from the digital wallet application to the merchant's server. The contents of the payload differ by wallet provider, but typically contain the DPAN, cryptogram, amount, and information about the cardholder (for example, name, etc.).

Payment Container

Element name for payment details that are securely passed from Meta Checkout to the merchant's server. The contents of the payment container typically contain the DPAN, cryptogram, amount and information about the cardholder.

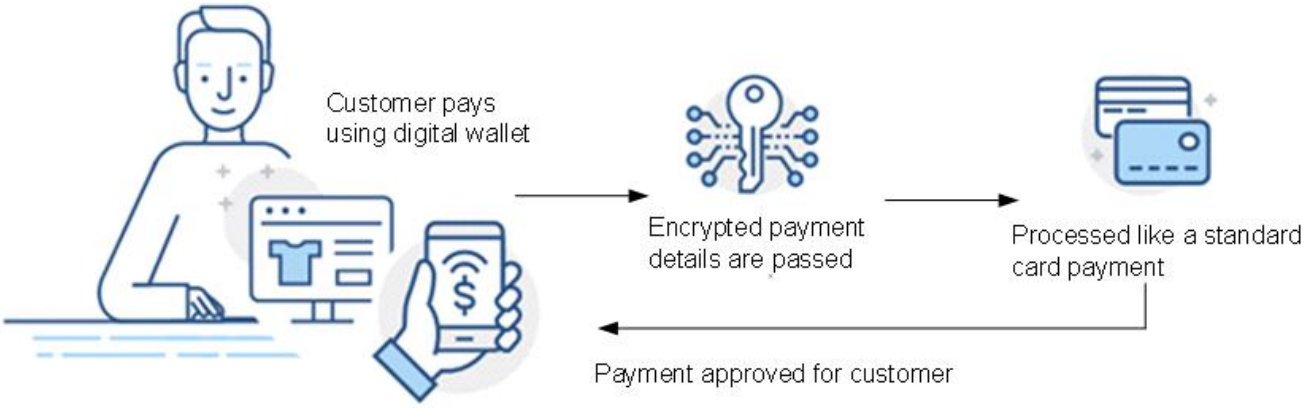
Payment Context

Meta Checkout data that uniquely identifies the context that the payment container is intended for. Meta encrypts container data by using the partner's public key before sending it to the merchant.

Overview

J.P. Morgan offers a simplified integrations with Apple Pay, Google Pay and Meta Checkout via the Digital Wallet API for in-app and e-commerce. The digital wallet service Application Programming Interface (API) is a unified framework that offers merchants ease of integration and flexibility in enabling only those digital wallets merchants prefer. The J.P. Morgan Orbital Gateway supports Authorization and Authorization and Capture processing for payment bundles originating from supported digital wallets. Additionally, J.P. Morgan offers a Debundle Only service, which returns the unencrypted transaction payload, and is then sent for processing to any of the J.P. Morgan host systems.

The Digital Wallet Application Programming Interface (API) can be used by merchants using Orbital, Tandem or Stratus. Merchants using Orbital can leverage one call authorization or the Authorization and Capture or Mark for Capture (MFC) APIs. Merchants using Tandem or Stratus should leverage the Debundle Only API and submit the Authorization, or Authorization and Capture, or Deposit Request message using the Consumer Digital Payment Token (CDPT) format in the Tandem and Stratus specification documents.



Onboarding

All merchants must adhere to the following onboarding requirements:

1. Establish a Merchant Services acquiring relationship and contract.
2. Establish an Orbital account.
3. Establish an account with the digital wallet provider (Apple Pay, Google Pay, and/or Meta Checkout.

4. Build a user interface within their app or e-commerce site to support digital wallet checkout flows.
5. Contact a J.P. Morgan Relationship Manager or Account Executive to schedule a digital wallet consultation.
6. The Relationship Manager or Account Executive assigns a Technical Implementations Analyst to review technical specifications, as well as discuss development/testing requirements with the merchant.
7. Credentials and questionnaires are provided to the merchant by the Technical Implementations Analyst to begin unattended testing.
8. The Technical Implementation Analyst provides a series of test cases in order to complete the integration testing.
9. Once all tests pass validation, the merchant has completed integration testing and an Integration Testing Summary is issued. J.P. Morgan proceeds with establishing production credentials.
10. The Digital Wallet AP implementation is complete. The merchant can accept digital wallet payments online or in-app.

Supported methods of payment

J.P. Morgan can process the following card schemes in the customer's digital wallet:

- Visa (VI)
- Mastercard (MC)
- American Express (AX)
- Discover/Discover Diners (DI)
- Japan Credit Bureau (JCB)

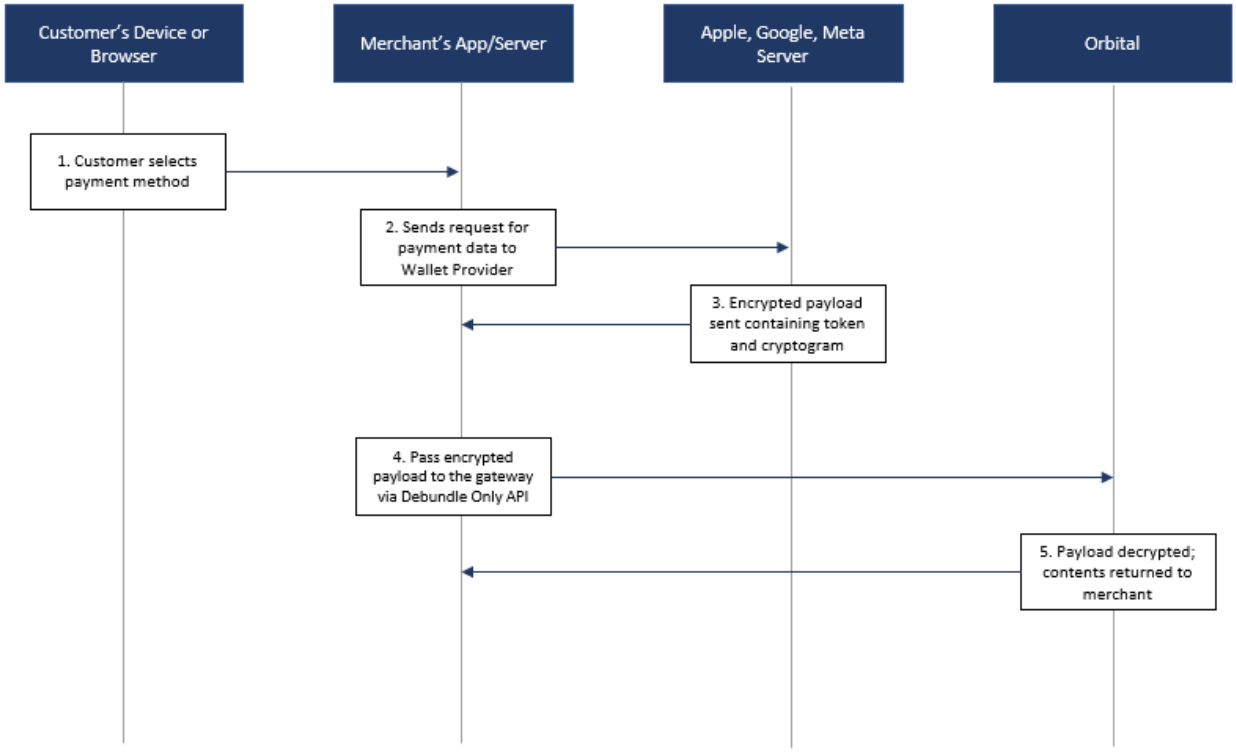
  Note: Stratus supports JCB transactions for merchants in the U.S. and Europe (EU). Tandem supports JCB transactions for merchants in the U.S.

How Do APIs work?

Debundle only

Support for Debundle only transactions are available for all digital wallets on the API service. For Apple Pay and Google Pay, the hosting merchant application interfaces with their digital wallet provider to obtain the encrypted payment bundle. The encrypted payment bundle is then sent to the merchant's server. Using JavaScript Object Notation (JSON) format, the merchant's server can send the bundle to the digital wallet service for decryption, or both decryption and processing. Decrypting only leverages a private key and public key pair, of which J.P. Morgan maintains the private key. The merchant sends the encrypted payment bundle to J.P. Morgan via the digital wallet. The decrypted payment bundle is then returned to the merchant.

Apple Pay, Google Pay, and Meta Checkout



1. The consumer selects either the **Apple Pay**, **Google Pay, or Meta Checkout** button as the method of payment from the merchant's application.

2. The merchant's application sends a request for payment data to the wallet provider's server.

3. The wallet provider’s server sends the encrypted payload, which contains data elements, such as the primary account number (Digital Primary Account Number [DPAN] or Funding Primary Account Number [FPAN]) and cryptogram, to the merchant’s server.

4. The merchant's server passes the encrypted payload to the Orbital Gateway via the Debundle only API.

5. The Orbital Gateway decrypts the payload and returns the contents to the merchant’s server.

Authorization and Authorization and Capture

Authorization and Authorization and Capture support are available for all digital wallets. For authorizations that are not marked for capture, additional integration testing is required in order to mark the transactions for capture at a later time.

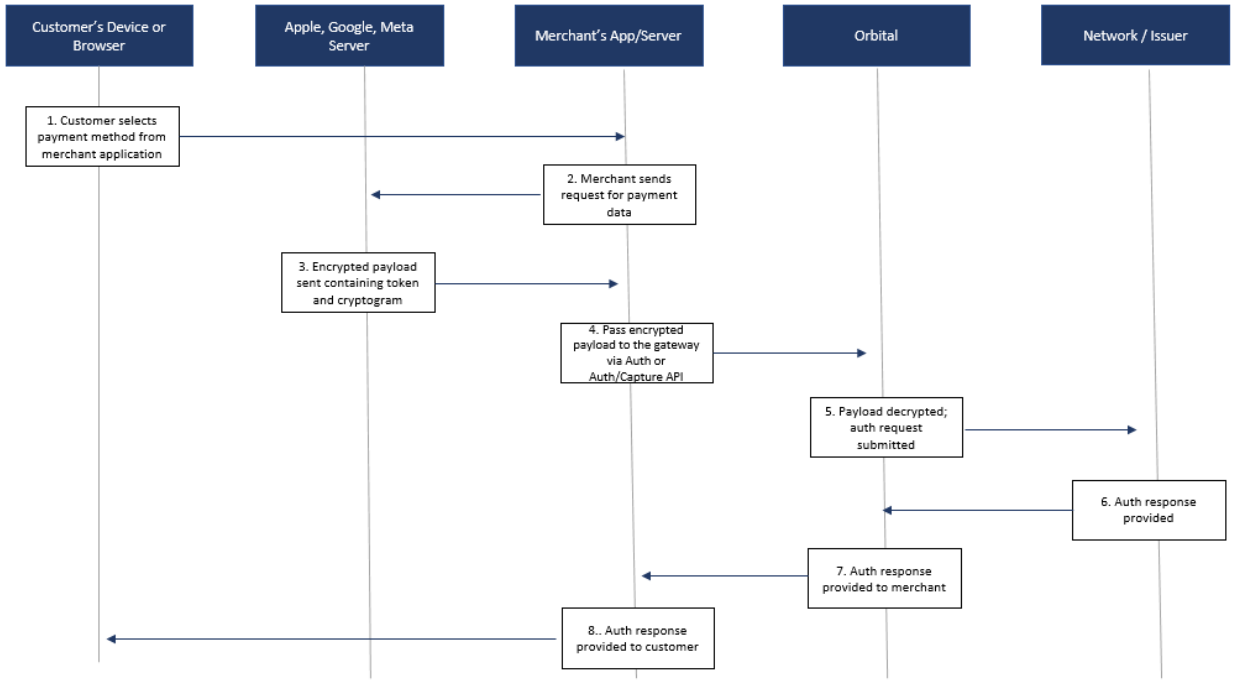
For Apple Pay, Google Pay, and Meta Checkout, the hosting merchant application interfaces with the digital wallet provider to obtain the encrypted payment bundle. The encrypted payment bundle is then sent to the merchant’s server. Using JSON format, the merchant’s server can send the bundle to the digital wallet service for decrypting and authorizing or decrypting, authorizing, and marking for capture.

Apple leverages a private key and public key pair, of which J.P. Morgan maintains the private key, and the merchant provides the public key within the payment bundle. Google and Meta leverage a private and public key pair at the processor level, while merchants are not required to provide a key in the payment bundle. Validation of the payment bundle is completed by a check on the Merchant Identifier (MID) or associated chain ID.

Orbital will attempt authorization using the encrypted payment details and the response is returned to the merchant’s server in JSON format.

Note: Using the CIT/MIT framework at the time of initial authorization enables the Orbital Gateway to perform re-authorizations of DPAN transactions (including when original authorizations have expired), allowing merchants to successfully perform capture requests.

Apple Pay, Google Pay, and Meta Checkout



1. From the merchant application, the consumer selects the **Apple Pay**, **Google Pay or Meta Checkout** button as the method of payment.

2. The merchant's application sends a request for payment data to the wallet provider's server.

3. The wallet provider's server sends the encrypted payload, which contains data elements, such as the primary account number (DPAN or FPAN) and cryptogram, to the merchant's server.

4. The merchant's server passes the encrypted payload to the Orbital Gateway via the Authorization or Authorization and Capture API.

5. The Orbital Gateway decrypts the payload and sends the authorization or authorization and capture request to the network/issuer.

6. The network/issuer approves or declines the authorization or authorization and capture request, and then sends a response to the Orbital Gateway.

7. The Orbital Gateway forwards the response to the merchant's server.

8. The merchant's server sends the Authorization or Authorization and Capture approval or decline response to the merchant's customer.

Mark for Capture

A Mark for Capture (MFC) is used to mark a previously authorized transaction as being ready to be submitted for clearing and is present for future fulfillment models. A transaction can be authorized at the present time and marked for capture at any time within a four month span following the time of transaction.

Caution: Authorization of certain payment options will age off after a number of days. The authorization time frames windows are up to the issuer, but general guidance for the different card brands are as follows:

- Visa, Mastercard, International Maestro, Amex = 7 days
- Discover = 10 days (all industries and MCC's; except for international card sales/credits and travel related MCC's including auto rental, airline, and passenger railway age off after 30 days).
- Note: For direct Amex relationships (conveyed) the client should obtain this information from Amex directly.

For Tandem/PNS (BIN 000002) merchants only: Orbital Gateway will perform an automatic re-authorization at the time of settlement if the authorization has aged off.

For Stratus (BIN 000001) merchants only: All transactions are subject to the host's re-authorization rules. The Stratus host may perform a re-authorization at the time of settlement, depending upon the merchant's parameters. Contact Merchant Services for additional information.

The MFC can be for an amount less than, equal to, or greater than the original authorization.

If the amount is less than the original authorization, it is processed in the same manner as a split transaction. This transaction also results in the creation of a new order for the remaining balance from the original authorization. Adjustments to the original transaction, such as level 2 data or the amount, are also made, as required. When marking a portion for capture, the system automatically attempts to obtain a new authorization for the remaining balance. This authorization attempt is performed for both BIN 000001 and BIN 000002 merchants.

Over-capture (i.e., capture amount greater than authorization) is available for specific industries, and is supported with approval. The merchant must work with their J.P. Morgan Relationship Manager to enable the over-capture. Refer to the **Processing and Interchange Guidelines** for industry tolerances by brand.

Refer to MFC request and response elements.

Digital Wallet API integration steps

Perform the steps below to implement the Digital Wallet AP.

Integration with Apple Pay

1. Create an Apple Developer's account.

   Note: Enter the following URL into an internet browser:

   https://help.apple.com/developer-account/#/dev2b5e6d209

2. Create a certificate.
3. Register the primary and sub-domains.
4. Create an MID and internet security certificate.
5. Conduct sandbox testing.

Note:

- Apple provides test cards for merchants to use during card brand testing.
- J.P. Morgan allocates Certificate Signing Requests (CSR) for each Apple MID. The merchant is required to put the public key on the Apple Developer's portal.

Integration with Google Pay

1. Create a Google developer's account.

   Note: Type the following URL into an internet browser:

   https://developers.google.com/pay/api

2. Select a processing method:

   - Processor/Gateway for Orbital
   - Direct merchant for Tandem or Stratus

3. Follow the Google Pay brand guidelines.

4. Complete the tutorial and integration checklist.

5. Conduct sandbox testing.

Note:

- Google Pay returns FPANs (i.e., browser-based transactions) and DPANs (i.e., device-based transactions), depending on how the transaction is initiated by the customer. Through the Google Pay setup, merchants can elect to accept only DPANs (Cryptogram_3DS) in order to limit the Payment Card Industry (PCI) scope.

- Google does not provide test cards. The Google Pay service generates secure payment bundles for the test environment with test account numbers. However, to generate test transactions, clients/developers must load personal account numbers to trigger the creation of secure payment bundles during test mode.

- Testing limitations imposed by Google prevents merchants from performing full-brand integration testing. At this time, integration testing is limited to format-only reviews.

- Merchants using the gateway Google Pay setup are not responsible for key management. However, merchants using the direct merchant Google Pay setup will receive a public key to be stored on the Google Developer's portal.

- When using Orbital to debundle a payload, select the **ECv1** or **ECv2** option.

Integration with Meta Checkout

1. Review Meta Checkout's acceptable use policy: https://transparency.fb.com/policies/other-policies/meta-pay-terms-and-conditions/
2. Chase will onboard you as a merchant on Meta Checkout platform
3. Integrate Meta Checkout JavaScript SDK in your checkout page: https://developers.facebook.com/docs/meta-pay/javascript
4. Conduct sandbox testing

Use case overview

The typical business use cases for digital wallets and their associated APIs are listed in the table below.

Use Case	Description/Purpose	API Name(s) or Endpoint
Authorization and Capture	A single transaction request with no follow-up needed. This action completes the transaction request process for settlement.	CWS Authorization and Capture
Authorization	A single transaction, which leaves the request open to be settled later or perform other actions against it.	CWS Authorization
Mark for Capture	Completes authorization only transactions, and is collected into settlement at the automatic settling time.	CWS MFC
Debundle Only	A request to decrypt the payload. Orbital returns the elements necessary to perform a transaction.	CWS Debundle
Profile Create	Stores the contents of the payload in an Orbital profile to be used in a future transaction.	CWS Add Profile

Use Case	Description/Purpose	API Name(s) or Endpoint
Profile and Authorization	Stores the details of the encrypted payload for later use, and issues an immediate authorization request.	CWS Add Profile + CWS MFC
Authorization, Capture with a Different Amount	Authorizes a transaction for the purchase amount, and captures for a higher amount.	CWS Authorization + CWS MFC
Partial Shipment	Authorizes a transaction for the full amount. Submits an MFC for the partial amount being shipped.	CWS Authorization + CWS MFC
Recurring Transactions	Submits an authorization for payments with a fixed or variable frequency. Authorization can be either merchant- or Chase-managed using the managed billing feature.	CWS Authorization + CWS MFC CWS Add Profile and CWS Managed Billing (optional)
Incremental Authorization	Submits an estimated authorization amount, as well as a subsequent incremented authorization amount to increase the total authorized amount. Submits a Mark for Capture when no additional increment is necessary, and is ready for deposit.	CWS Authorization + CWS MFC

Note: Refunds and reversals/voids should be executed using Virtual Terminal (VT).

Integration testing

All merchants and integrators must complete integration testing before going live. Merchants are able to use the Digital Wallet Application Programming Interface (API) integration testing environment to fully test their implementation. Following a successful testing but the merchant, a J.P. Morgan analyst will also test the integration. For additional information regarding integration testing, consult with a Merchant Services Technical Implementation Manager.

This section outlines where to locate information on the integration testing environment, as well as its related testing phases and resources available to support the testing process.

Additional assistance

Product support is available through Merchant Services Gateway Support at 1-(866) 645-1314, or via email at gatewaysupport@chasepaymentech.com. Merchants must to provide a merchant/division number, username, and company name.

Digital Wallet API testing environment

The Digital Wallet API testing environment allows merchants to perform integrated testing of their Digital Wallet API implementation against a code base that mirrors the J.P. Morgan production environment. This API testing environment allows merchants to test the following integrations:

- Digital Wallet Debundle Only API
- Digital Wallet Authorization API
- Digital Wallet Authorization and Capture API
- Digital Wallet Mark for Capture (MFC) API
- Digital Wallet Create (add) Profile API
- Digital Wallet Update Profile API
- Digital Wallet Fetch (retrieve) Profile API

While there are two phases of testing (merchant testing and integration testing), both phases leverage the same integration testing environment.

Testing environment setup

Prior to testing, several integration points must be connected to the digital wallet integration testing environment. A J.P. Morgan Technical Implementations Manager is able assist with questions.

Note: Google Pay test pay bundles expire in 7 days.

Test cases

Contact a Chase Technical Implementation Manager for test cases.

Notifications

Apple Pay

Automated Key Expiring Emails to Merchants using Apple Pay for Debundle Only:

- The key management process for merchants using the Apple Pay Debundle Only service is managed by merchants.
- A notification is sent once the Apple Pay Certificate Signing Request (CSR) expires following each 24-month interval.
- Automated emails are to the Relationship Manager, technical contact, and primary contact associated with the merchant or chain setup.
- An email will be sent at 30-, 15-, and 5-day increments, as well as 24 hours, prior to key expiration.
- The email contains the last 4 digits of the merchant or chain ID in the header, and the Apple Merchant Identifier (MID) in the body. This provides merchants with instructions for uploading new CSRs to the Apple Portal.

Google Pay

Automated Key Expiring Emails to Merchants using Google Pay for Debundle Only:

- The key management process for merchants using the Google Pay Debundle Only service is managed by the merchants.
- A notification is sent once the Google Pay public key expires (i.e., every 12 months).
- Automated emails are sent to the Relationship Manager, technical contact, and the primary contact associated with the merchant or chain setup.

- Emails will be sent during 30-, 15-, and 5-day increments, as well as 24 hours, prior to key expiration.
- The email contains the last 4 digits of the merchant or chain ID in the header, and the Google MID in the body. This provides merchants with instructions for uploading new public keys to the Google Developer's portal.

Functional processing

Tokenization

Digital wallet transactions, with the exception of Google Pay Funding Primary Account Number (FPAN) transactions, are tokenized according to Europay, Mastercard and Visa Co., LLC (EMVCo) standards. The credit card's Primary Account Number (PAN) is replaced with a token or Digital Primary Account Number (DPAN). Once a customer submits a payment, the token is submitted for payment in place of the PAN, along with a cryptogram and other applicable information in order to securely authenticate the transaction.

Cryptograms

Digital wallet e-commerce transactions require the use of cryptograms. Certain card brands have specific requirements for where the cryptogram should be passed in an authorization message.

- Visa (VI) - **CAVV** field
- Mastercard (MC) - **DSRP** field
- American Express (AX) – **AEVV** field
- Discover (DI) – **CAVV** field
- Japan Credit Bureau (JCB) – **Digital Token Cryptogram** field (**CAVV** field)

  Note: A valid value in the `targetCardBrand` field is required for JCB DPAN transactions in the U.S. and EU.

  - For BIN 000001 EU merchants only, JCB DPAN transactions are processed on the Visa network and use the same elements as Visa.
  - For U.S. merchants, JCB DPAN transactions are processed on the Discover network and use the same elements as Discover.

Recurring payments

A recurring payment scenario occurs when a merchant obtains a consumer's explicit approval to be charged on a recurring basis. Typical examples of recurring payments include utility and subscriptions services. VI, MC, DI, and AX all require the use of their Cardholder Initiated Transaction (CIT) and Merchant Initiated Transaction (MIT) framework for new implementations that support recurring transactions. Refer to the specifications of your acquiring platform for more information regarding how to use this framework.

In a recurring payment scenario:

- The first transaction is formatted as a regular single transaction.
- Subsequent transactions will use **ECI=2** for recurring transactions.

CIT and MIT framework

In addition to the transaction generated by a cardholder-initiated event, a significant segment of transactions exist where a merchant, Payment Facilitator (PF), or Staged Digital Wallet Operator (SDWO) uses a cardholder's payment credentials (i.e., account details) for future purchases. Stored payment credentials include information stored by a merchant or its agent, a payment facilitator, or a staged digital wallet operator to process future transactions. This stored information includes, but is not limited to, account numbers and/or payment tokens.

With the introduction of the stored credential and MIT framework, data is presented with the authorizations necessary to identify those stored credentials, as well as indicate whether cardholder consent has been obtained. Within these frameworks, transactions are presented as either a CIT or MIT.

The CIT/MIT feature is supported for VI, MC, AX and DI. The CIT/MIT feature for JCB and UnionPay transactions are limited to those that process over the Discover network, and use the same elements as Discover.

Note:

Within the framework, merchants are responsible for receiving and retaining the Transaction ID (TXID) for use during subsequent transactions.

Using the CIT/MIT framework for initial authorizations enables the Orbital Gateway to perform re-authorizations of DPAN transactions (including when the original authorization has expired), allowing merchants to successfully settle delayed or partial captures.

Soft descriptors

Soft descriptor records define a merchant's name or product that may appear on a consumer's statement. Soft descriptor data is optional. Merchants may take advantage of Merchant Services' soft descriptor record specifications to submit the merchant's **Name** and/or **Product Description** fields, whichever is most recognizable to the consumer. Additionally, soft descriptor records provide merchants with greater flexibility in describing the consumer's purchase. The **Merchant City/Customer Service Phone Number** field allows merchants to identify a business location, or to provide consumers with a customer service phone number or URL. The soft descriptor data is then submitted and passed to the card association (along with the transaction), and then posted on the consumer's statement, if applicable.

Soft descriptors are supported for AX, ChaseNet, DI, Discover Diners, International Maestro (IM), JCB, MC, MC Canadian Domestic Restricted Debit, VI, and VI Canadian Domestic Restricted Debit.

Support for soft descriptors is not globally available to all customers using the Orbital Gateway, and the behavior differs from downstream host platforms. Refer to the **Stratus Specifications** on Developer Center for additional information regarding authorization and settlement. Note: Tandem/Paymentech Network Services (PNS) supports soft descriptors with restrictions. Refer to PNS (BIN 000002) Support.

Merchant Services risk/credit department approval is required prior to sending soft descriptors. The merchant must also be set up to send soft descriptor records or transactions containing this information will be declined. It is subject to the issuer's discretion whether this descriptor will be displayed on the cardholder's statement.

The table below describes the generic soft descriptor terms used throughout this document, along with their respective JavaScript Object Notation (JSON) element names.

Soft Descriptors – Field Names

Field Name	JSON
Merchant Name	softDescMercName
Product Description	softDescProdDesc

Field Name	JSON
Merchant City	softDescMercCity
Customer Service Phone Number	softDescMercPhone
Merchant URL	softDescMercURL
Merchant Email Address	softDescMercEmail

Note: Although some soft descriptor records can be populated in any given combination, all soft descriptor elements must be submitted in the transaction request. Any unpopulated elements should be null-filled.

Stratus (BIN 000001) support

The Orbital Gateway supports soft descriptors into the Stratus host for authorization and settlement. However, merchants must adhere to the following requirements:

- Prior risk department approval is required.
- The merchant/terminal ID must be enabled for soft descriptors on the Orbital Gateway.

Refer to the **Stratus Specifications** on Developer Center for additional information.

Rules and guidelines – credit card

Merchant Services does not generate or segregate reports by soft descriptor. If the merchant wishes to render Stratus reports segregated by product, specific reporting divisions must be established and deposited under that division number.

For merchants who wish to combine several merchant names under one corporation, contact a Merchant Services representative for details regarding the use and regulation of soft descriptors.

The description in the **Merchant Name** field should be the most recognizable to the cardholder, and should consist of the company and/or trade name, as well as a description of the purchased product or service. The merchant name can be one of three different lengths:

- 3 bytes
- 7 bytes
- 12 bytes

The product description can be appended, based on the length of the merchant's name, such that they are a combined length of 21 bytes. Additional options include the following:

- 18 bytes
- 14 bytes
- 9 bytes

Note:

- The **Merchant City** field allows merchants to identify business locations, or to provide cardholders with a customer service phone number or URL. This field is a requirement in order to qualify for VI's lowest direct marketing interchange rate.
- If a merchant submits a backslash (\) in the merchant descriptor, it is converted to a hyphen (-) on the cardholder's statement. If a merchant submits a question mark (?) in the merchant descriptor, it is converted to a space on the cardholder's statement.
- Certain AX card types/programs ignore descriptors sent using soft descriptors, such as the Optima card. Merchants should contact the appropriate AX representative for additional information.
- Non-Ecommerce transactions sent with a URL do not qualify for the best interchange.
- For MC, Mail Orders/Telephone Orders (MOTO) and recurring industry types, if the **City/Phone** field at the division level is not a customer service phone number, one must be populated. Failure to do so results in a response reason code of **BP**, which is an error code indicating customer service phone numbers are required for MOTO and recurring MC transactions only.
- The Orbital Gateway will apply the asterisks (*) in the necessary locations. Do not add these to a request.

Tandem/PNS (BIN 000002) support

The Orbital Gateway supports soft descriptors into the Tandem (i.e., PNS) host. However, the following exceptions must be taken into account:

- Only Canadian Merchant Services customers are supported.
- The merchant/terminal ID must be enabled for soft descriptors on the Orbital Gateway.
- The behavior differs from that of the Stratus interface. Refer to the **PNS Specifications** manual for additional information.
- Unlike Stratus, the only value passed on to the cardholder's statement is the **Merchant Name** field, which, for these customers, is a maximum of 25 bytes. All other soft descriptor fields can be sent optionally, but will not be submitted to the settlement host, nor will it display on the cardholder's statement.

Note: Contact a Merchant Services representative for setup information for either host.

Soft Descriptor examples

Example 1: Soft descriptor section for a 3-byte merchant descriptor with phone number:

```
softDescMercName = XYZ
softDescProdDesc = PAYMENT1OF3
softDescMercCity =
softDescMercPhone = 888-888-8888
softDescMercURL =
softDescMercEmail =
```

Example 2: Soft descriptor section for a 12-byte merchant descriptor with email

```
softDescMercName = XYZCOMPANY
softDescProdDesc = PYMT1OF3
softDescMercCity =
softDescMercPhone =
softDescMercURL =
softDescMercEmail = suppt@xyz.com
```

Note: The **Phone**, **URL**, and **Email** fields contain a maximum of 13 characters. Therefore, care should be taken when supplying this data so that consumers are able to understand the information on their statements.

Profile Management

The Orbital Gateway includes a Profile Management functionality, allowing customer payment information to be stored within Orbital. This information can be accessed by a custom profile ID (i.e., token) to process future transactions. Merchants are able to process transactions by passing a token value representing a cardholder. Using the customer profile ID (i.e., token) simplifies transaction processing, as well as mitigates any data entry errors. Additionally, using a customer profile ID eliminates the need to store sensitive information in-house, so merchants are free to focus on their business objectives, while Merchant Services focuses on securely processing the transactions.

Once a profile is created, transactions can be processed, using either the online interface or the Orbital Virtual Terminal (VT), by simply referencing customer profiles and populating any additional information not stored in those profiles. This feature is only available to merchants using the Orbital Gateway interface.

Managed billing extends the capabilities of profiles to include recurring, installment and deferred billing. Using this feature, merchants can configure Orbital Gateway to initiate payments on a desired future date.

Merchant accounts must be enabled for the Profile Management tool in order to utilize the managed billing functionality. Contact Merchant Services if a merchant has not already set up the program. Once enabled, the next step is to create customer profiles using following methods:

- Create a profile as a distinct action.
- Create a profile as part of a transaction request.

Once a profile exists, it can be used to process new transactions. The information stored in the profile is used to populate transaction data elements. Merchants have the option to override any part of the profile for subsequent transactions. Profiles can be updated or deleted at any time.

Information Saved in a Profile

Whether a profile is created during a profile add transaction, added on-the-fly during an authorization transaction, or updated at a later time during a profile update transaction, the following list defines which profile data elements can be saved:

- Customer Reference Number
- Required and Not Editable (i.e., profile ID)

- Customer Name
- Customer Email

  Note: Only available for profile add or update transactions. This value is not yet available for on-the-fly profile adds within authorization transactions.

- Address Information:
  - Address 1
  - Address 2
  - City
  - State
  - Zip Code
  - AVS Country Code
  - Phone
- Amount
- Order Description
  - Able to be set in the following ways:
    - By sending a specific description message in the **comments** tag.
    - By setting the **profileOrderOverideInd** to populate the **comments** tag.
- Order ID
  - Accomplished by setting the **profileOrderOverideIn** to populate the **orderID** tag.
- Payment Information
  - Credit Card
    - Card Number
    - Expiration Date

- Note: Profile data remains static, unless it is changed by a merchant-initiated profile update request.

Information Not Saved in a Profile

There are a number of data elements that are not added to a profile, regardless of how it is entered, including, but not limited to, the following:

- Level 2 Data
- Card Verification Number (e.g., Card Verification Value [CVV] 2, Card Verification Code [CVC] 2, and Card Identifier [CID]).
    - Card association rules forbid the storing of card verification number information. It must be requested from a cardholder on a case-by-case basis.
- VI Secure and MC Identity Check data

Transaction Types

Profiles may be used on the following transaction types:

- Authorization
- Authorization and Capture
- Prior Authorizations
- Refund
- SafeTech Fraud Analysis

Profile usage is not functional or necessary for the following:

- Voids/Reversals
- Mark for Capture (MFC)
- End of Day

Industry Types

Industry types supported by the Orbital Gateway (e.g., e-commerce, mail order, recurring, and interactive voice response) are supported within each profile.

Currencies

All currencies supported by the Orbital Gateway are supported as a part of each profile.

Managed billing

Managed billing enables merchants to configure profiles so that Merchant Services is able to automatically run future transactions. Additionally, managed billing supports recurring, installment, and deferred billings. A billing schedule can be set to start on a certain date, follow a weekly, monthly, or yearly recurrence pattern, and optionally end on a certain date, or once a specified number of billings has elapsed. Orbital automatically processes transactions on behalf of the merchant in accordance with the preset schedule.

Managed Billing Profiles

A merchant account can only be configured for one type of managed billing at a time.

Recurring Billings

Recurring billings bill cardholders for future payments according to a pre-defined schedule. Recurring billings can be configured to occur on a weekly, monthly, and/or yearly basis. Attributes such as start date, end date and recurring frequency must be established so that the managed billing system is able to schedule payments. Since Merchant Services initiates future transactions, a choice must be made regarding order ID generation.

Installment Billings

Installment billings are processed exactly like recurring billings, with the exception that the end billings trigger is configured using the **mbRecurringMaxBillings** tag. However, this behavior is not enforced by the Orbital Gateway.

Deferred Billings

Deferred billings are one-time billings that occur on a pre-determined date. The key element that needs to be established for a deferred billing is the deferred billing date.

As with recurring billings, Merchant Services initiates the future transaction instead of the merchant. Therefore, a choice must be made regarding order ID generation. Refer the **Orbital Gateway Web Service Interface Developer Guide** for additional information regarding managed billing settings.

Incremental authorization

Incremental authorization allows merchants to increase the total amount of authorized funds once additional products or services are added to a customer's order. This feature is available as part of the CIT/MIT framework within the **CEST** and **MINC** MIT type codes. Incremental authorization is supported for VI, ChaseNet, MC, and IM card brands. Refer to the **Orbital Gateway Web Service Interface Developer Guide** for additional information regarding Incremental Authorization.

Level 2 processing

Level 2 processing data fields are used in business-to-business environments. Merchants have the ability to collect funds in conjunction with the settlement of procurement credit card transactions.

The Orbital Gateway supports the processing of procurement cards, including the enhanced data required by various card associations.

- Stratus and Tandem (i.e., PNS) merchants:
    - VI, MC and DI
    - For Stratus merchants:
        - AX level 2 and Transaction Advice Addenda (TAA)

Note: Level 2 data sets were initially supported for the subset of procurement cards known as "purchasing cards." Orbital Gateway has expanded to include a superset of procurement cards known as "commercial cards." Purchasing and commercial cards should not vary with respect to level 2 requirements. To maintain support of legacy integrations, level 2 data elements are referenced in this API as "purchasing card data."

BIN ranges

Bank Identification Number (BIN) ranges are assigned by card associations, and identify purchasing and/or commercial cards. BIN ranges are subject to change at the card associations' discretion.

Processing

Level 2 data can be sent either with the original (via Authorization or Authorization and Capture) request, or appended to the transaction via an MFC request, if not originally supplied in the authorization request.

Level 2 data can be sent with sales and refunds for both Stratus and Tandem (i.e., PNS) merchants.

MFC adjustment of Level 2 data

Level 2 data is supplied either on Authorization or MFC requests.

Functional processing

- During settlement, the Orbital Gateway uses the data presented with an Authorization request.
    - Level 2 data is submitted with both the authorization and an MFC request for the full amount of the authorization.
- The data submitted with an MFC supersedes that of the authorization in its entirety.
    - Level 2 data is submitted with both the authorization and an MFC request for a partial amount of the authorization.
    - A split transaction is generated. By default, the data submitted in the first MFC is used on all subsequent splits. Each additional MFC may supersede this data with relevant level 2 data, if desired.

- Level 2 data is only submitted with the MFC.
    - During settlement, the Orbital Gateway uses the data presented with an MFC request.
    - If the amount of the MFC request is less than the authorized amount, a split transaction is generated. By default, the data submitted in the first MFC request is used on all subsequent splits. Each additional MFC request may supersede this data with relevant level 2 data, if desired.

Additional information

Each card brand has subtle data requirement differences in the order to properly qualify for level 2 transactions. There are also a few differences in data formats between the Stratus and Tandem (i.e., PNS) hosts.

Virtual Terminal

All functionality supported through this interface for level 2 data is additionally available through the Orbital Gateway Virtual Terminal (VT).

Processing interface description

The digital wallet service uses JavaScript Object Notation (JSON) to manage requests using Hypertext Transfer Protocol Secure (HTTPS).

Endpoint environments

Merchant Services exposes redundant hostname/port network endpoints to ensure high availability for digital wallet services. To ensure maximum availability, developers should code to detect connectivity issues and Hypertext Transfer Protocol (HTTP) errors, and temporarily switch to a failover Uniform Resource Locator (URL). Failovers to the secondary hostname/port must be automatic and completely transparent to the end-user. Communication with the primary hostname/port should be attempted periodically while in a failover state.

Caching IP addresses of Merchant Services servers is strongly discouraged. For load balancing and redundancy reasons, the digital wallet service processing is divided amongst multiple data centers. Therefore, the DNS service should be used to determine the destination IP address for each transaction.

While the integration testing environment is available for testing at all hours, it is only monitored for availability during business hours (8:00 AM – 5:00 PM Eastern Standard Time [EST], Monday – Friday). Additionally, the hardware in place is designed primarily for integration testing, and not load testing. If there is a need to ensure uptime outside of normal business hours, consult a J.P. Morgan Integration Testing Analyst.

Digital Wallet service URLs

Debundle Only Endpoints

Orbital Gateway Integration Testing System

- Primary: https://orbitalvar1.chasepaymentech.com/cws/1/debundle/api
- Secondary: https://orbitalvar2.chasepaymentech.com/cws/1/debundle/api

Orbital Gateway Production System

- Primary: https://orbital1.chasepaymentech.com/cws/1/debundle/api
- Secondary: https://orbital2.chasepaymentech.com/cws/1/debundle/api

Authorization endpoints

Orbital Gateway Integration Testing System

- Primary: https://orbitalvar1.chasepaymentech.com/cws/1/auth
- Secondary: https://orbitalvar2.chasepaymentech.com/cws/1/auth

Orbital Gateway Production System

- Primary: https://orbital1.chasepaymentech.com/cws/1/auth
- Secondary: https://orbital2.chasepaymentech.com/cws/1/auth

Authorization and Capture endpoints

Orbital Gateway Integration Testing System

- Primary: https://orbitalvar1.chasepaymentech.com/cws/1/authcap
- Secondary: https://orbitalvar2.chasepaymentech.com/cws/1/authcap

Orbital Gateway Production System

- Primary: https://orbital1.chasepaymentech.com/cws/1/authcap
- Secondary: https://orbital2.chasepaymentech.com/cws/1/authcap

Mark for Capture URLs

Orbital Gateway Integration Testing System

- Primary: https://orbitalvar1.chasepaymentech.com/gwapi/1/gateway/markforcapture
- Secondary: https://orbitalvar2.chasepaymentech.com/gwapi/1/gateway/markforcapture

Orbital Gateway Production System

- Primary: https://orbital1.chasepaymentech.com/gwapi/1/gateway/markforcapture
- Secondary: https://orbital2.chasepaymentech.com/gwapi/1/gateway/markforcapture

Profile management URLs

Create (add) profile

Orbital Gateway Integration Testing System

- Primary: https://orbitalvar1.chasepaymentech.com/gwapi/1/gateway/profile/add
- Secondary: https://orbitalvar2.chasepaymentech.com/gwapi/1/gateway/profile/add

Orbital Gateway Production System

- Primary: https://orbital1.chasepaymentech.com/gwapi/1/gateway/profile/add
- Secondary: https://orbital2.chasepaymentech.com/gwapi/1/gateway/profile/add

Update profile

Orbital Gateway Integration Testing System

- Primary: https://orbitalvar1.chasepaymentech.com/gwapi/1/gateway/profile/change
- Secondary: https://orbitalvar2.chasepaymentech.com/gwapi/1/gateway/profile/change

Orbital Gateway Production System

- Primary: https://orbital1.chasepaymentech.com/gwapi/1/gateway/profile/change
- Secondary: https://orbital2.chasepaymentech.com/gwapi/1/gateway/profile/change

Fetch (retrieve) profile

Orbital Gateway Integration Testing System

- Primary: https://orbitalvar1.chasepaymentech.com/gwapi/1/gateway/profile/fetch

- Secondary: https://orbitalvar2.chasepaymentech.com/gwapi/1/gateway/profile/fetch

Orbital Gateway Production System

- Primary: https://orbital1.chasepaymentech.com/gwapi/1/gateway/profile/fetch
- Secondary: https://orbital2.chasepaymentech.com/gwapi/1/gateway/profile/fetch

Security

Given the inherent risks associated with processing transactions over the internet, the Digital Wallet Service requires both of the following:

- Encrypted traffic to prevent interception of a payload
- Authentication of the source request generation.

The sub-sections below define how the system manages that security.

Secure sockets layer implementation requirement

The digital wallet service Uniform Resource Locator (URL) must be accessed using Hypertext Transfer Protocol – Secure (HTTPS) so that private information is securely transferred. This requires merchants to use a Secure Sockets Layer (SSL) implementation, which are available for most programming languages. Merchants are responsible for gaining the necessary expertise to open a secure channel for the service.

Interfacing with the digital wallet service using SSL does not require merchants to have a certificate. Digital wallet services use a non-authenticated SSL session, meaning that merchants are not authenticated using a digital certificate as a component of the SSL negotiation. Refer to the Authentication section for additional information regarding how Merchant Services authenticates merchant traffic.

Non-SSL postings should never be made across an external or unsecured network. If a clear text request is made to one of the digital wallet service URLs, the service will return an error condition (an **HTTP 403** error), along with the accompanying JSON payload containing a **ProcStatus 20403** error.

Authentication

The digital wallet service supports connection username/password/Merchant Identifier (MID) authentication for incoming requests. This means the username, password and MID are passed in the message header. Each must match what is registered on the Merchant Services servers in order to process transactions in the integration testing or production environments.

An HTTP 412 error is returned for all activity wherein the connection username/password/MID is not registered on the Merchant Services servers. The accompanying JSON payload contains a ProcStatus 20412 error. Additionally, the connection username must be affiliated with the client's MID for the following reasons:

Third-party hosting service organizations are able to present on behalf of other merchants to submit transactions. However, each time a new customer is added, the merchant or third-party hosting organization must ensure that new MIDs or chain IDs are affiliated with the hosting company's connection username.

If merchants expect to have more than one merchant account on the system, it could have its connection username affiliated at the chain-level hierarchy within the system.

Each time a new MID is added, it will function properly as long as it is placed within the same chain. If it is not placed within the same chain, the additional MIDs must be affiliated with the connection username. For example, J.P. Morgan generally affiliates all Stratus accounts (BIN 000001) with their company number, so all MIDs or divisions under that company are automatically affiliated.

Google Pay authentication includes a validation of the MID or chain ID supplied within the payment bundle. Connection credentials should match the level of the setting for the username and password. For example, if a Google Pay chain ID is used within the Google Pay API, then a chain-level username and password should be used.

- Merchants that process on the Orbital Gateway can leverage the same connection username and password for their digital wallet service requests.

MID-Association Failures

If a connection username is registered, but the merchant presents an MID that is **not** been associated with the username, the digital wallet service will return a **ProcStatus 20412** error.

Connection username/password/MID format

The connection username and password must be registered on Merchant Services servers. Each is submitted within the message header, under the following corresponding elements:

- OrbitalConnectionUsername
- OrbitalConnectionPassword
- MerchantID

The connection username and password must follow specific formatting rules. Both username and password are subject to the following requirements:

- Must be between 8–32 characters.
- Must contain at least one number.
- Must contain only standard English letters or digits (a-z, A-Z, 0-9).
- Cannot contain embedded spaces.

Connection passwords are **case-sensitive**, while connection usernames are not. If additional information is needed, contact a Technical Implementation Analyst or Account Executive.

A MID must adhere to the following requirements:

- BIN 000001: 6-digit Stratus division number
- BIN 000002: 12-digit Tandem/Paymentech Network Services (PNS) MID
- Field type: Numeric (maximum of 12 characters)

For Existing Merchants Using IP-based Authentication:

Internet Protocol (IP)-based authentication and connection username/password authentication are exclusive to one other. If a merchant is set up for both IP-based and connection username/password authentications, request messages are authenticated based on whether connection username and/or password elements exist within the payload.

If either element exists, the digital wallet service will attempt to validate the username/password values. If the authentication fails, the digital wallet service will not revert to an IP-based authentication.

Message specifications

Communication protocol

The digital wallet service only supports Hypertext Transfer Protocol Secure (HTTPS) communication. This method provides a single-threaded (i.e., synchronous) model, in which a merchant creates an HTTPS request to the service, and then blocks until the service sends back the HTTPS response. While HTTPS requests are single-threaded, a single interface is able to create multiple simultaneous HTTPS requests.

Posting to a URL

The digital wallet service provides responses only to HTTP Point of Sales Terminal (POST) requests. The POST method is used to request that the origin server accepts the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line. The digital wallet service does not support Graph Editor Toolkit (GET) requests.

JSON schema

The digital wallet service accepts JSON requests, and then returns JSON responses defined by Merchant Services. The JSON format to interface with the digital wallet service is given in sample JSON transactions (refer to Google Pay transactions, Apple Pay transactions, Meta Checkout transactions, and Debundle Only samples).

Errors or unexpected behaviors can result if any characters in the request payload do not match the character encoding specified in the request.

Summary of API calls

Headers

MIME header

Multipurpose Internet Mail Extensions (MIMEs) are mechanisms for specifying and describing the format of internet message bodies. The digital wallet service supports both `HTTP/1.0` and `HTTP/1.1` MIME header specifications for describing the message payload, along with supporting information that allows it to process incoming transaction requests, as well as their outgoing replies.

Request MIME-Header Definition

The table below lists elements within the MIME-Header for all Apple Pay and Google Pay digital wallet requests. The values associated with each element are case sensitive, but the element names themselves are not.

Element Name	Description	Required	Max Char	Field Type
MIME-Version	Always `1.0` or `1.1`	O	X.X	N
Content-type	Always `application/json`	M	Var	A
Interface-Version	Optional `MIME-Header` element that can be used by Merchant Services in production support.	O	Var	AN
Accept	Always `application/json`	M	Var	A

Element Name	Description	Required	Max Char	Field Type
OrbitalConnectionUsername	**Orbital connection username** User name set up on the digital wallet service. • Between 8–32 characters (a-z, A-Z, 0-9) • Minimum of 1 number • No leading, trailing or embedded spaces • Not case-sensitive	M	32	AN
OrbitalConnectionPassword	**Orbital connection password** Password used in conjunction with connection username. • Between 8–32 characters (a-z, A-Z, 0-9) • Minimum of 1 number • No leading, trailing, or embedded spaces • Case-sensitive, and must match what is stored on the system.	M	32	AN

Element Name	Description	Required	Max Char	Field Type
MerchantID	**Merchant ID** BIN 000001: 6-digit Stratus Division Number BIN 000002: 12-digit Tandem/Paymentech Network Services (PNS) Merchant Identifier (MID) Can be used in a multi-merchant chain setup and is subject to validation checks on Orbital Gateway.	M	12	N

POST/AUTHORIZE HTTP/1.1 Element

Although the request line is not part of the `MIME-Header` element, this value is static and should always be presented as `POST/AUTHORIZE HTTP/1.1`.

HTTP Post: The digital wallet service only provides responses to HTTP Point of Sales Terminal (POST) requests.

Authorization and Authorization and Capture Message

Request elements – Authorization and Authorization and Capture

The table below lists elements for an Authorization and Authorization and Capture requests.

Note: Key value pairs are case-sensitive for specific values.

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
audit	N/A	**Audit** The high-level parent element for the location and device from which the transaction originated. Used by the Orbital Gateway to validate the authenticity of the transaction for its approval. Geo-coordinates can be sent to the gateway as an optional value for validation.	M	N/A	N/A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
latitudeLongitude	audit	**Latitude and Longitude** Coordinates of the device making a request. Note: Geo-location data is optional. If not participating in geolocation, Default to **1**,**1**. Format: ±180.99999,±180.99999	M	21	N
politicalTimeZone	audit	**Political Time Zone** Format: Use Coordinated Universal Time (UTC) offset. For example, EST during daylight savings (EDT) is represented as `-0400`. EST during the remainder of the year would be represented as `-0500`. Space-fill if the time zone is not included, or is shorter than full length.	O	5	AN
vendorId	audit	**Vendor ID** Value assigned by Merchant Services to identify the application vendor. Required when `softwareID` is provided.	C	4	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
softwareId	audit	**Software ID** Value assigned by Merchant Services to identify the application type and version. Required when `vendorID` is provided.	C	4	AN
mobileDeviceType	audit	**Mobile Device Type** A description of the type of mobile device used by the customer. Valid values: • 80 = iPhone • 40 = iPod Touch • 20 = iPad • 10 = Android • 08 = Blackberry • 04 = Other	O	2	N
encryptedPaymentBundle	N/A	**Encrypted Payment Bundle** The content of the encrypted payment bundle should be in JavaScript Object Notation (JSON) format, as received from Google or Apple.	M	Var	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
paymentContainer	encryptedPaymentBundle	**Payment Container** Contains the payment instrument that is required to process the user's payment as received from Meta Checkout.	C	Var	AN
paymentContext	N/A	**Payment Context** To ensure security in transit, Meta encrypts container data by using the partner's public key before sending it to the merchant.	C	Var	AN
billAddress	N/A	**Billing Address** The high-level parent element for the billing address of the card used for the transaction. The `billAddress` fields are used in the AVS check.	O	N/A	N/A
name	billAddress	**Cardholder Billing Name**	O	30	A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
address1	billAddress	**Cardholder Billing Address Line 1** Should not include %. BIN 000001 merchants must supply address1, city and zip in order for data to be transmitted to the host processing system.	O	30	AN
address2	billAddress	**Cardholder Billing Address Line 2** Should not include %	O	30	AN
city	billAddress	**Cardholder Billing City** Should not include % BIN 000001 merchants must supply address1, city and zip in order for data to be transmitted to the host processing system.	O	20	A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
state	billAddress	**Cardholder Billing State** Should not include any of the following characters: • % (percent) • | (vertical slash) • ^ (caret) • \ (backward slash) • / (forward slash)	O	2	A
zip	billAddress	**Cardholder Billing Address Zip Code** All AVS requests must include a 5-digit zip code. If sending a zip code + 4, separate with a hyphen (-). BIN 000001 merchants must supply address1, city and zip in order for data to be transmitted to the host processing system.	O	10	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
countryCode	billAddress	**Cardholder Billing Address Country Code** Valid values: • US = United States • CA = Canada • GB = Great Britain • UK = United Kingdom This field should be left blank for all other countries.	O	2	A
phone	billAddress	**Cardholder Billing Phone Number** Format: AAAEEENNNNXXXX, where: • AAA = Area Code • EEE = Exchange • NNNN = Number • XXXX = Extension Example: 9998887777	O	14	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
phoneType	billAddress	**Customer Telephone Type** Valid values: • D = Day • H = Home • N = Night • W = Work The default is **H** if any phone number is present, and this element is either missing or null-filled.	O	1	A
orderId	N/A	**Merchant-Defined Order Number** A unique identification value of an order generated by the hosting application. For a single order, there could be one or more transaction reference numbers associated due to partial approvals.	M	22	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
comments	N/A	**Free-form Comments** Comments that can be entered by a merchant. The information is stored with the transaction details. For Tandem (i.e., PNS) customers, this field populates in the **Customer Defined Data** field, which is displayed in Resource Online.	O	256	AN
cardIndicators	N/A	**Enhanced Authorization: Card Type Indicators** Available to BIN 000001 merchants to request additional response information. This value is ignored on unsupported transactions. Valid values: • Y = Card indicators should be returned, if available. • N = Card indicators should not be returned.	O	1	A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
partialAuthInd	N/A	**Partial Authorization Support Indicator** Used to indicate to the card issuer if the host application can support the logic necessary to perform and manage a partial approval response. For a partial approval, business logic must be placed into the host application to determine what action should be taken with the balance owed. Valid values: • Y = Specify the issuer should return a partial authorization if needed. • N = Specify the issuer should not return a partial authorization. • S = Stratus (BIN 000001) only; Indicates a partial authorization can be supported without attempting to override host settings. Supported for VI, Mastercard (MC), American Express (AX), and Discover (DI) only.	O	1	A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
		Note: PINless Debit e-commerce supports partial authorizations.			
walletType	N/A	**Wallet type used for making the request** Valid values: • 1 = Apple Pay • 2 = Google Pay • 4 = Meta Checkout	M	1	N
transactionAmount	N/A	**Amount of Transaction** Implied decimal, including those currencies that are a zero exponent. For example, both $100.00 (an exponent of 2) and ¥100 (an exponent of 0) should be sent as an amount of **10000**. Required element for Google Pay. For Apple Pay, the transaction amount is received as a part an encrypted payment bundle. This field is be used to accept transaction amount.	C	12	N

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
bin	N/A	**Bank Identification Number (BIN)** Transaction Routing Definition Assigned by Merchant Services. • 000001 = Stratus • 000002 = Tandem (PNS) Required for Google Pay and Meta Checkout, but is optional for Apple Pay.	C	6	N

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
targetCardBrand	N/A	**Target Card Brand** Used by Japan Credit Bureau (JCB) digital wallet transactions to indicate the card brand processing the transaction. Required for JCB Digital Primary Account Number (DPAN) transactions in the U.S. and Europe (EU). Valid values: • DI = DI (US) • VI = VI (EU) **Notes:** • For BIN 000001 EU merchants only, JCB DPAN transactions are processed on the VI network, and therefore, use the same elements as VI. • For U.S. merchants, JCB DPAN transactions are processed on the DI network, and therefore, use the same elements as DI.	C	2	A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
softDescriptor	N/A	**Soft Descriptor parent element** The high-level parent element for the merchant soft descriptor used for the transaction. Refer to Soft Descriptors.	O	N/A	N/A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
merchantName	softDescriptor	**Soft Descriptor Merchant Name** • Conditionally required for soft descriptors. • The **Merchant Name** field should be the most recognizable to the cardholder (e.g., company name or trade name). The actual length of the **Merchant Name** field is conditionally tied to host, as well as the size of the `productDesc` field used. **Stratus**: Credit – Three options that conditionally affect the `productDesc` are as follows: • Maximum of 3 bytes • Maximum of 7 bytes • Maximum of 12 bytes	C	25	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
productDesc	softDescriptor	**Soft Descriptor Product Description** Conditionally required for soft descriptors. Provides an accurate description. **Stratus**: Credit: • If `softDescMercName` = 3 bytes (maximum of 18) • If `softDescMercName` = 7 bytes (maximum of 14) • If `softDescMercName` = 12 bytes (maximum of 9)	C	18	AN
merchantCity	softDescriptor	**Soft Descriptor Merchant City** Tag conditionally required for soft descriptors. Merchant city for retail. Field is required, but should be null-filled if any soft descriptor data is submitted.	C	13	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
custServicePhone	softDescriptor	**Soft Descriptor Merchant Phone** Field conditionally are required for soft descriptors. Only one of the location soft descriptor values should be sent (e.g., phone, URL, or email). All others should be null-filled. This field does not display on cardholder statements for Tandem (i.e., PNS) merchants. Valid formats include the following: • NNN-NNN-NNNN • NNN-AAAAAAA	C	12	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
merchantURL	softDescriptor	**Soft Descriptor Merchant URL** Field conditionally are required for soft descriptors. Only one of the location soft descriptor values should be sent (e.g., phone, URL, or email). All others should be null-filled. This field does not display on cardholder statements for Tandem (i.e., PNS) merchants.	C	13	AN
merchantEmail	softDescriptor	**Soft Descriptor Merchant Email** Field conditionally required for soft descriptors. Only one of the location soft descriptor values should be sent (e.g., phone, URL, or email). All others should be null-filled. This field does not display on cardholder statements for Tandem (i.e., PNS) merchants.	C	13	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
mit	N/A	**CIT/MIT parent element** The high-level parent element for processing CIT/MIT transactions.	C	N/A	N/A
mitMsgType	mit	**CIT/MIT Message Code** Indicates the message type used for the message type records. Examples include the following: • CSTO/CGEN/ CINS/CUSE/CREC/CREV/CEST for customer-initiated codes. • MUSE/MINS/MRAU/MREC/MREV/MRSB/MINC/MNOS/MDEL for MITs.	C	4	A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
mitSubmittedTransactionID	mit	**MIT Submitted Transaction ID** Submitted CIT/MIT Transaction Identifier (TXID) in the request The submitted TXID returned to the merchant from a previous authorization request in a series of transactions. The TXID is not sent for CIT transactions, but is a required value for MIT transactions. Note: This tag is not required for merchants/clients that use Orbital's Profile Management service. This tag is only required for merchants/clients who store and manage their customer's payment credentials outside of Orbital.	C	15	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
mitStoredCredentialInd	mit	**Stored Credential Flag** Indicates that the cardholder's credentials are on-file with the merchant. Valid values: • Y – The cardholder's credentials are on-file with the merchant • N – The cardholder's credentials are not on-file with the merchant " " – Blank	O	1	A
addProfileFromOrder	N/A	**Add Profile from Order** Method to use to generate the customer profile number. When creating a profile during an order request, this tag defines how the customer profile number is generated. Valid values: • A = Auto-generate the `customerRefNum` • S = Use `customerRefNum` field • O = Use `orderID` as the `customerRefNum` • D = Use comments as the `customerRefNum`	C	1	A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
profileOrderOverideInd	N/A	**Profile Order Override Indicator** Defines if any order data can be pre-populated from the customer reference number (`customerRefNum`). Conditionally required when adding a profile as part of an authorization. Valid values: • NO = No mapping to order data • OI = Use `customerRefNum` for `orderID` • OD = Use `customerRefNum` for comments • OA = Use `customerRefNum` for `orderID` and `comments` Field must be empty (or null-filled, if including the element) when using a profile during an authorization request. Alternatively, if not being used, this field can be excluded from the request.	C	2	A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
customerRefNum	N/A	**Customer Reference Number** Sets the customer reference number that utilizes a customer profile on future orders. Required if the `customerProfileFromOrderInd` option is **S** (use the `customerRefNum` field). If `customerProfileFromOrderInd` is **A**, the customer reference number is defined by the Orbital Gateway, and any value passed in this element is ignored. Given this value can be the same as the order number, valid characters for this field follow the same conventions as the order ID element, and include the following: • Alphabetical (lower- and upper-cased) • Numeric (0 – 9) • - , $, @, **&**, and a space character	C	22	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
		Notes: • A space character cannot be the leading character. • An ampersand must be sent as `&` • Double ampersands (`&&`) are not supported. All alphabetic characters in this field are stored as uppercase by the Orbital Gateway. Uppercase and lowercase values cannot be used to differentiate customer reference numbers.			

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
digitalTokenCryptogram	N/A	**Digital Token Cryptogram** The gateway will parse any data provided in this field and convert it into the proper format. **For Discover CDPT transactions:** • This number must be base 64 encoded. • Cryptographic value derived with an algorithm that applies the issuer's private key to the combination of the cardholder account number, transaction identifier and other data. **For Consumer Digital Payment Tokens:** • Unique transaction cryptogram generated by the digital wallet provider, and should be submitted as it was received. • Note: Merchants must use this field to submit a new Accountholder Authentication Value (AAV)/CAVV when a new authorization is needed.	C	120	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
managedBilling	N/A	**Managed Billing parent element** The high-level parent element for setting up and/or processing managed billing transactions	C	N/A	N/A
mbType	managedBilling	Managed Billing Type Indicates the type of managed billing in which the merchant is participating. Valid values: • R = Recurring • D = Deferred The value submitted must be in agreement with the type of managed billing for which the merchant is configured at Merchant Services. This field serves to notify the Orbital Gateway that the transaction is a managed billing transaction. If this field is not sent with a managed billing transaction, all other **Managed Billing** fields are ignored.	C	1	A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
mbOrderIdGenerationMethod	managedBilling	**Managed Billing Order ID Generation Method** This value is used to set the method used by Orbital to generate order IDs for managed billing transactions. This field does **not** influence the order ID for stand-alone transactions initiated by the merchant, VT transactions, etc. Valid values: IO = Use the customer reference number (profile ID). This value is made up of the capital letters I and O, and no numbers. DI = Dynamically generates the order ID. This value is made up of the capital letters D and I, and no numbers.	C	2	A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
mbRecurringStartDate	managedBilling	**Managed Billing Recurring Start Date** Defines the future date at which Orbital will begin a recurring billing cycle with the associated profile. To allow the managed billing engine to properly calculate and schedule all billings, this date must be at least one day following the request date. A recurring billing cycle cannot begin on the date at which the request message is sent to the Orbital Gateway. Format: MMDDYYYY	C	8	N
mbRecurringEndDate	managedBilling	**Managed Billing Recurring End Date** Defines the future date at which Orbital will end a recurring billing cycle with the associated profile. Format: MMDDYYYY This is the first of three possible recurring end triggers. Only one end trigger may be submitted per request message.	C	8	N

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
mbRecurringNoEndDateFlag	managedBilling	**Managed Billing No End Date Indicator** Valid values: • Y = Schedule recurring transactions for an infinite amount of time. A Y in this field overrides the value, if any, in the mbRecurringEndDate field. • N (or blank) = Orbital will use the value of the `mbRecurringEndDate` field to define the recurring end date. This is the second of three possible recurring end triggers. Only one end trigger can be submitted per request message.	C	1	A
mbRecurringMaxBillings	managedBilling	**Managed Billing Maximum Number of Billings** This value defines the maximum number of billings that will be allowed for a recurring billing cycle. Valid values: 1–999999 This is the third of three possible recurring end triggers. Only one end trigger can be submitted per request message.	C	6	N

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
mbRecurringFrequency	managedBilling	**Managed Billing Recurring Frequency Pattern** This pattern is a subset of a standard Command Run On (CRON) expression, comprising 3 fields separated by white space. Fields: • Day of month • Month • Day of week Permitted values: • 1-31 (for day of month) • 1-12 (for January - December) • 1-7 (for day of week) Permitted characters: • , - * ? / L W (for day of month) • , - * / (for month) • , - * **? / L #** (for day of week)	C	64	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
mbDeferredBillDate	managedBilling	**Managed Billing Deferred Billing Date** Defines the future date at which Orbital will trigger a one-time billing with the associated profile. This date must be at least one day following the request date. A deferred billing cannot take place on the date at which the request message is sent to the Orbital Gateway. Format: MMDDYYYY	C	8	N

M = Mandatory, C = Conditional, O = Optional, N/A = Not Applicable

Refer to the following sections for Authorization and Authorization and Capture request samples:

- Google Pay
- Apple Pay
- Meta Checkout

Response elements – Authorization and Authorization and Capture

The table below lists the elements for an Authorization and Authorization and Capture response.

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
ResponseCode	N/A	**Response Code parent element** The high-level parent element for response.	M	N/A	N/A
authorizationCode	ResponseCode	**Issuer Approval Code** Unique transactional-level code issued by the bank or service establishment for approval.	C	6	AN
visaVbVRespCode	ResponseCode	**Cardholder Authentication Verification Value (CAVV)** Response code used for VI secure transactions. Conditional on CAVV value being sent.	C	1	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
procStatus	ResponseCode	**Process Status** The first data set that should be checked to determine the result of a request. The only element that is returned in all response scenarios. Identifies whether transactions have successfully passed all Gateway edit checks (0 – Success)	M	6	AN
procStatusMessage	ResponseCode	**Process Status Message** Text message associated with the `respCode` value	M	Var	A
approvalStatus	ResponseCode	**Approval Status** • 0 = Declined • 1 = Approved • 2 = Message/system error	M	1	N

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
respCode	ResponseCode	**Response Code** Normalized authorization response code issued by the host system (Stratus/Tandem [i.e., PNS]), which identifies either an approval (**00**), or the reason for a decline or error.	M	2	AN
respCodeMessage	ResponseCode	**Response Code Message** Text message associated with `respCode` value	C	Var	A
hostResponseCode	ResponseCode	**Actual Host Response Code** Exact response sent by the host authorization system (non-normalized by the Gateway). For systems that have already coded to the Stratus/Tandem (i.e., PNS) authorization response values, they are available via this field.	M	3	A
hostResponseCodeMessage	ResponseCode	**Host Response Code Message** Text message associated with `hostResponseCode` value.	C	Var	A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
avsRespCode	ResponseCode	**AVS Response Code** Address Verification Request Response Conditional on AVS request being sent.	C	2	AN
hostAvsRespCode	ResponseCode	**Actual Host Address Verification Response Code** Exact address verification response sent by host authorization system (non-normalized by the Gateway). For those systems that have already coded to the Stratus/Tandem authorization response values, they are available via this field.	M	2	AN
EnhancedAuth	N/A	**Enhanced Authorization** The high-level parent element for enhanced authorization response.	O	N/A	N/A
ctiAffluentCard	EnhancedAuth	**Card Indicator: Affluent Category** Affluent cards have very high preset spending limits, if any. Returned only for BIN 000001 merchants on applicable transactions.	O	1	A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
ctiCommercialCard	EnhancedAuth	**Card Indicator: Commercial Card** Returned only for BIN 000001 merchants on applicable transactions.	O	1	A
ctiDurbinExemption	EnhancedAuth	**Card Indicator: Durbin** Returned only for BIN 000001 merchants on applicable transactions.	O	1	A
ctiHealthcareCard	EnhancedAuth	**Card Indicator: Healthcare Card** Returned only for BIN 000001 merchants on applicable transactions.	O	1	A
ctiLevel3Eligible	EnhancedAuth	**Card Indicator: Level 3 Data Eligibility** Returned only for BIN 000001 merchants on applicable transactions. • Y=Yes • N=No • null	O	1	A
ctiPayrollCard	EnhancedAuth	**Card Indicator: Payroll Card** Returned only for BIN 000001 merchants on applicable transactions.	O	1	A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
ctiPrepaidCard	EnhancedAuth	**Card Indicator: Prepaid Card** Returned only for BIN 000001 merchants on applicable transactions.	O	1	A
ctiPINlessDebitCard	EnhancedAuth	**Card Indicator: PIN-less Debit Eligibility** Returned only for BIN 000001 merchants on applicable transactions.	O	1	A
ctiSignatureDebitCard	EnhancedAuth	**Card Indicator: Signature Debit Eligibility** The term “signature debit” refers to the processing of a debit card as a credit card. Returned only for BIN 000001 merchants on applicable transactions.	O	1	A
ctiIssuingCountry	EnhancedAuth	**Card Indicator: Issuing Country** Used to distinguish a customer as either domestic or international. Format: 3-character International Organization for Standardization (ISO) country code. Returned only for BIN 000001 merchants on applicable transactions.	O	3	A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
ctiPrepaidReloadableCard	EnhancedAuth	**Card Type Indicator (CTI) Prepaid Reloadable Card** Indicates whether the card is prepaid reloadable, non-reloadable or an anonymous card. Valid values : • Y=reloadable • N=non-reloadable • A=Anonymous • X=Not Applicable/Unknown Notes: • Returned only for BIN 000001 merchants on applicable transactions. • Returned only if WSDL version=4.4 or higher is sent in the request.	O	1	AN
txRefNum	N/A	**Gateway Transaction Reference Number** A unique value for each transaction, and is required in order to adjust any transaction in the Orbital Gateway (e.g., Mark for Capture [MFC] or Void [V]).	M	40	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
remainingBalance	N/A	**Remaining Card Balance** Indicates the amount remaining on a card when returned in the response from the issuer.	C	Var	N
requestedAmount	N/A	**Requested Transaction Amount** Echoes the amount from the request.	C	12	N
redeemedAmount	N/A	**Redeemed Transaction Amount** Indicates the amount returned in the response from the host.	C	12	N
partialAuthOccurred	N/A	**Partial Authorization Occurred** Indicates if a partial approval was returned. This tag will be null if a partial approval has been returned.	C	1	A
hash1	N/A	Internal reference number	N/A	N/A	N/A
hash2	N/A	Internal reference number	N/A	N/A	N/A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
lastFourFPAN	N/A	**Last four digits of FPAN** Last four digits of FPAN will be sent back if request contains FPAN.	C	4	N
lastFourDPAN	N/A	**Last four digits of DPAN** Last four digits of DPAN will be sent back if request contains DPAN.	C	4	N
orderId	N/A	**Merchant-Defined Order Number** Field defined and supplied by the authorization originator, and echoed back in response.	M	22	AN
cardBrand	N/A	**Card Type/Brand for the Transaction** Returns the card type/brand as processed on the host platform.	M	2	A
profileProcStatus	ResponseCode	**Result Status of Profile Management** Communicates the success or failure of a profile management request: • 0 = Success • >0 = An error condition (refer to Error Handling: Profiles for values).	M	6	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
profileProcStatusMsg	ResponseCode	**Profile Process Status Message** Verbose text description associated with `profileProcStatus`.	M	Var	A
mcRecurringAdvCode	ResponseCode	**Recurring Payment Advice Code** Valid values: • 01 = New account information available. Obtain new account information. • 02 = Try again later. Recycle transaction in 72 hours. • 03 = Do not try again. Obtain another type of payment from customer. • 24 = Retry after 1 hour • 25 = Retry after 24 hours • 26 = Retry after 2 days • 27 = Retry after 4 days • 28 = Retry after 6 days • 29 = Retry after 8 days • 30 = Retry after 10 days Note: MC recurring transactions only.	M	2	N

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
mitReceivedTransactionID	N/A	**Received TXID** The received TXID returned to the merchant. This field will always have a value for CIT/MIT transactions.	C	15	AN
customerName	N/A	**Customer Billing Name** Echoes the customer name sent in the request.	O	30	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
customerRefNum	N/A	**Customer Reference Number** Sets the customer reference number that will be used to utilize a customer profile on all future orders. Required if the `customerProfileFromOrderInd` option is **S** (use the `customerRefNum` field). If `customerProfileFromOrderInd` is **A**, the customer reference number will be defined by the Orbital Gateway, and any value passed in this element will be ignored. Given that this value can be the same as the order number, the valid characters for this field follows the same convention as the order ID element, and includes: • Alphabetical (lower- and upper-cased) • Numeric (0 – 9) • **-** , **$**, @, **&**, and a space character	C	22	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
		Notes: • A space character cannot be the leading character. • An ampersand must be sent as `&` • Double ampersands (`&&`) are not supported. All alphabetic characters in this field are stored as uppercase by the Orbital Gateway. Uppercase and lowercase values cannot be used to differentiate customer reference numbers.			

M = Mandatory, C = Conditional, O = Optional, N/A = Not Applicable

Refer to the following sections for Authorization and Authorization and Capture response samples:

- Google Pay
- Apple Pay
- Meta Checkout

Authorization and Authorization and Capture samples

Request sample – Authorization and Authorization and Capture – Google Pay


```
{
"audit": {
    "latitudeLongitude": "1,1",
    "politicalTimeZone": "0500",
    "mobileDeviceType": "80"
    },
"encryptedPaymentBundle":
{"signature":"MEQCID7me9PEtUNcra0pjwi5YLTx6J0AL/Yzc1s0aDIy85VQAiAmwHJexjH9J8UkvHS/SlfX
IatAa3vkQq/kYWBFGN7Lcg\u003d\u003d","protocolVersion":"ECv2","signedMessage":"{\"encry
ptedMessage\":\"TO8TPNwf7+gGlXvgg8i9b99b299kIpUVLOKRVFRIq4evzIg8TbE9qY4gMKOHshy446STxo
3FS1IAqA6hC9h/Q1EcT8nXrnYhymek0Cv1NcESC7r5Z7vvF10w9KPXOlYHZoiz2yEeJDEm4f0F9v6XYUfw4J6G
TvWZ/lYOXNv6j9D5855T+1ED7sXIZshzHofz9UbGLTb+/g2f8QpVzINQlW9dIQ9HmDsNXTT9ID2s/SgdM7+wUy
MRgSF746HuLZjQjVX7gV4Ag3EWqnl+FaJaMYKo5mDawGr0IVobWFiLHtEt7YVxvoV8+i9mdI9MESTFmiKZ99Vu
y1eUoA6hsO8vHRnrnu3kyePuIvzQ7DkElprJ2EYiC17Ix+R4YzXDO9l1TQUHTKhozS2HLL17t/Nho+B0GWSgsL
XZHEPCaRmBkozt9D2gCvt03b5YRiYtsBmn0A\\u003d\\u003d\",\"ephemeralPublicKey\":\"BIrp+aB8
f0Kpyymylyyns3+tHGwxlSiGnLP2unSWTcocg/EfCaFFbhOVMHHikI4Uv1kYCR66gX9KluAOQwA1/zM\\u003d
\",\"tag\":\"/9QIu+rwIEUr/9td5TE5u5pzEJ3HZwfZymOAJVm7fyY\\u003d\"}"
    },
"billAddress" : {
    "name": "Billing Address Name",
    "address1": "Billing Address 1",
    "address2": "Billing Address 2",
    "city": "Billing Address City",
    "state": "GA",
    "zip": "33711-4444",
    "countryCode": "US",
    "phone": "9998887777",
    "phoneType": "W"
    },
"orderId": "123456",
"comments": "Comments",
"cardIndicators": "Y",
"partialAuthInd": "Y",
"bin": "000001",
"transactionAmount": 1000,
"walletType": "2"
}
```

Response Sample – Authorization and Authorization and Capture – Google Pay


```
{
"ResponseCode": {
    "authorizationCode": "tst007",
    "visaVbVRespCode": "",
    "procStatus": "0",
    "procStatusMessage": null,
    "approvalStatus": "1",
    "respCode": "00",
    "respCodeMessage": "Approved",
    "hostResponseCode": "100",
    "hostResponseCodeMessage": null,
    "avsRespCode": "3 ",
    "hostAvsRespCode": " "
},
"EnhancedAuth": {
    "ctiAffluentCard": "N",
    "ctiCommercialCard": "N",
    "ctiDurbinExemption": "Y",
    "ctiHealthcareCard": "N",
    "ctiLevel3Eligible": "N",
    "ctiPayrollCard": "X",
    "ctiPrepaidCard": "N",
    "ctiPINlessDebitCard": "N",
    "ctiSignatureDebitCard": "N",
    "ctiIssuingCountry": "USA"
},
"txRefNum": "5AD9DF5F7F91BB04241FC162CCD2A881C5E05365",
    "remainingBalance": null,
    "requestedAmount": 1000,
    "redeemedAmount": 1000,
    "partialAuthOccurred": null,
    "hash1": "3adfe2a59aeede7b3d090c3b2293bb1e07c79e8015da69372c1d3d55078de29b",
    "hash2": "a46014fle3a3018d44e5c5eecb80a7027cf8bce743eec713d7d8fe89fbf3778c",
    "lastFourFPAN": "1111",
    "lastFourDPAN": null,
    "orderId": "123456",
    "cardBrand": "VI"
}
```


Generating Encrypted/Signed Payload for Google Pay

Merchants must pass two parameters to the Google Pay API in order to generate the encrypted/signed payment bundle, as defined in the table below.

Google Pay Parameter	Description
gateway	Specific value used for processing Google Pay through the Orbital Gateway digital wallet service.

Google Pay Parameter	Description
gatewayMerchantId	Gateway merchant/chain account number assigned by Merchant Services. This account number will match that of the merchant host platform: • BIN 000001: 6-digit Stratus division number • BIN 000002: 12-digit Tandem (i.e., PNS) MID

Request Sample – Authorization and Authorization and Capture – Apple Pay


```
{
"audit": {
    "latitudeLongitude": "1,1",
    "politicalTimeZone": "0500",
    "mobileDeviceType": 80
},
"encryptedPaymentBundle": {
  "data" :
"IzxSm6YWehmlLvk5HY/rsl4hhWuorOG7R6ERP0fqzTokMhS5JtyAU8ajPIu/aHcbOxYQOhvk/K+3n6N7SbEKg
SuT100YFmeIKh3IkSLa4u1/1Y4Z9y5bqZFPxd8IcQnuR8HZKgJDHCXQzDDYP4JBMtqZQzRztzsIfa4eoOnGuZC
c2s+WxGap4iv92vPj8tAHonvSE9t0ByUCBLgfvu25GR0eJb6UM8nBvxP2/qBSElOuyLo80enrZ6tlp3xtpBEV8
oeOc9iLSmalayfD7JQxZXd2cWA/sZPWn4VGIj7Dt05NYE/iFZrw2VOa2hOJ4/4dOGS1KJzhw+RPRufhadAF96k
7O3LwbMphcM9sZLN/Y/LSqVFGzIq6ZlrnOwcxzvjNqw4ccNl4v3eehL4TRRgfF3LirV56BeADzJmq0pB3W/vu"
,
  "header": {
    "ephemeralPublicKey":
"MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEQ3CCwyRLUK61yxYifPLYY87iWcPydTCL0PpAOkpOAvZDCCffK
bQTsxK9707qmVrAmH0wDNZEbLJ9Ob3teiiCbA==",
    "publicKeyHash" : "MUwkjyUBpyRiZTVMUrIzA6+SIrr9mV8nNct6YO0rGNg=",
"transactionId" : "ad9256898767x9618750998af3058af5b8ede5fb67bbdc37370b289743f762a7"
  },
  "signature":
"MIAGCSqGSIb3DQEHAqCAMIACAQExDzANBglghkgBZQMEAgEFADCABgkqhkiG9w0BBwEAAKCAMIIBYjCCAQigA
wIBAgIGAV1lOPsBMAoGCCqGSM49BAMDMDgxITAfBgNVBAMMGFBheW1lbnRlY2ggTW9iaWxlIFNESyBDQTETMBE
GA1UECgwKUGF5bWVudGVjaDAeFw0xNzA3MjExMjU2NTlaFw0zNDA3MjExMjU2NTlaMDgxITAfBgNVBAMMGFBhe
W1lbnRlY2ggTW9iaWxlIFNESyBDQTETMBEGA1UECgwKUGF5bWVudGVjaDBZMBMGByqGSM49AgEGCCqGSM49AwE
HA0IABEKuXMH9Q3bZlekeTuImojxPuHQnxA4jIKiFwF3wOH6nQY94asOmLLLws3JD9tv2M2P7ppUl961r15aw4
8Gnr2UwCgYIKoZIzj0EAwMDSAAwRQIhAIvcLMW83wgdvH0Mhi1ZJa93CV5bY6Ru5GKY/0vNb1F4AiBO4bPOqW7
YR8GlJ6x823vx+AATTg5gocYGrj8tquPnjQAAMYIBGzCCARcCAQEwQjA4MSEwHwYDVQQDDBhQYXltZW50ZWNoI
E1vYmlsZSBTREsgQ0ExEzARBgNVBAoMClBheW1lbnRlY2gCBgFdZTj7ATANBglghkgBZQMEAgEFAKBpMBgGCSq
GSIb3DQEJAzELBgkqhkiG9w0BBwEwHAYJKoZIhvcNAQkFMQ8XDTE3MDkxODEyMzA1NVowLwYJKoZIhvcNAQkEM
SIEIFOTICKavR26ewV/9jepdbFWNoASpvLan5brcCutlZHzMAoGCCqGSM49BAMCBEgwRgIhAO8S85/SS1fX0TR
yDu7RA5wO/lRTF2ayk1PPcE9IN7i3AiEApAP4zETvW3jpipxp/nrKcISIGSm+XTmHXCiJZB/vthMAAAAAAAA="
,
  "version" : "EC_v1"
},
"billAddress" : {
    "name": "Billing Address Name",
    "address1": "Billing Address 1",
    "address2": "Billing Address 2",
    "city": "Billing Address City",
    "state": "GA",
    "zip": "33711-4444",
    "countryCode": "US",
```

```
    "phone": "9998887777",
    "phoneType": "W"
},
"orderId": "123456",
"comments": "Comments",
"cardIndicators": "Y",
"recurringInd": "",
"partialAuthInd": "Y",
"walletType": "1"
}
```


Response Sample – Authorization and Authorization and Capture – Apple Pay


```
{
"ResponseCode": {
    "authorizationCode": "tst434",
    "visaVbVRespCode": "A",
    "procStatus": "0",
    "procStatusMessage": null,
    "approvalStatus": "1",
    "respCode": "00",
    "respCodeMessage": "Approved",
    "hostResponseCode": "100",
    "hostResponseCodeMessage": null,
    "avsRespCode": "3 ",
    "hostAvsRespCode": " "
},
"EnhancedAuth": {
    "ctiAffluentCard": "N",
    "ctiCommercialCard": "N",
    "ctiDurbinExemption": "N",
    "ctiHealthcareCard": "Y",
    "ctiLevel3Eligible": "N",
    "ctiPayrollCard": "X",
    "ctiPrepaidCard": "N",
    "ctiPINlessDebitCard": "N",
    "ctiSignatureDebitCard": "N",
    "ctiIssuingCountry": "USA"
},
"txRefNum": "5C94ACF0F1B781FFE30AF65884B12BDDED6453D0",
"remainingBalance": null,
"requestedAmount": 1000,
"redeemedAmount": 1000,
"partialAuthOccurred": null,
"hash1": "3adfe2a59aeede7b3d090c3b2293bb1e07c79e8015da69372c1d3d55078de29b",
"hash2": "a46014fle3a3018d44e5c5eecb80a7027cf8bce743eec713d7d8fe89fbf3778c",
"lastFourFPAN": null,
"lastFourDPAN": "9990",
"orderId": "123456",
"cardBrand": "VI"
}
```


Request sample - Authorization and Authorization and Capture – Meta Checkout


```
{
"audit" : {
"latitudeLongitude" : "1,1",
```

```
"politicalTimeZone" : "0500",
"vendorId" : "00C0",
"softwareId" : "019A",
"mobileDeviceType" : 80
},
"encryptedPaymentBundle" : {
  "paymentContainer" :
"eyJhbGciOiJFUzI1NiIsIng1YyI6WyJNSUlDT0RDQ0FTQ2dBd0lCQWdJQkFEQU5CZ2txaGtpRzl3MEJBUXNGQ
URBZk1SMHdHd1lEVlFRRERCUkdRbEJoZVNCMFpYTjBJRzF2WkdVZ2NtOXZkREFlRncweU1ERXlNRFV3TXpFeE1
qZGFGdzB6TURFeU1ETXdNekV4TWpkYU1DY3hKVEFqQmdOVkJBTU1IRVpDVUdGNUlIUmxjM1FnYlc5a1pTQnBib
lJsY20xbFpHbGhkR1V3V1RBVEJnY3Foa2pPUFFJQkJnZ3Foa2pPUFFNQkJ3TkNBQVR6Y2hXM0ZTWUdWeCsyTHJ
iRmFVQWtIOEord2N3RE80QUtVMFRNRjZjV1ZlazVYQ3RqMmQ4b1BhK1d6QVd2UjA3bFJxUGJpSklUWFBUUjVMT
GxoRXFXbzBJd1FEQWRCZ05WSFE0RUZnUVUyNFYweFZ4bWNrT0o1d2l3eTl2Smx3eUxxb1V3SHdZRFZSMGpCQmd
3Rm9BVUtseXRHZXBhY3BnTXUwV2ZyM2FsTndlYnR3MHdEUVlKS29aSWh2Y05BUUVMQlFBRGdnRUJBS0ptRVpaM
HQwZ3NoNFd3V0xzdmFSeFAvYVFkc2V5dGY4dTRrT2xlVHpWdUUzTnk1WGRqeWNKcDVURS9UNVFFb21HbWRsWFV
FSmQvKzhNNU1JTnVxS1c3Zlh1RVlaNE9GSlBhY3dvT2JPYXlyTXFGWXBIZ2NUV2lKVjJZTXdraDNNRnF6L2w3c
VZ4LzF5dVlNampmMlVyMERXVTJLY1dZQ0NZcEpFNTk1V3M0b3ZHc3dwemFDcDVoWWs1OUhxbkZrald4U2U3dkc
4aUlseGE0NWF5WkdLSThKMTNBZ3V1b2RGWng4UWVUdm9PTE1paUVDZXE5MXJySXNIOFFiSkliK0pueU1RTGFHU
mFEZlYvMTdlYjBFQzJQbnlvYXV6ZFlibHRuOXIvNHAzSXpRRXpBVk5DS0tQRHdUMU85UnJ2bFVyYk11UmhyRE9
uZ2tqeXE3OWRZNS9JPSJdfQ.eyJhbW91bnQiOnsiY3VycmVuY3kiOiJVU0QiLCJ2YWx1ZSI6MTAwMDB9LCJjb2
50YWluZXJfZGF0YSI6ImV5SmxjR3NpT25zaWVTSTZJbTloV2paSWQyMVBkMVZxUjBWclMwVTBiRE0xUjJOM05U
QnJPVVJ2Ym5CeVpWSm9ZM2x3ZEROQ01IY2lMQ0o0SWpvaWVucHBUbUkwZUdoMlRUZDVkMVZ5ZFhOUGEwdEROMm
80VDFwR00xaGhWREUxYlZKMFlWY3pUbTFKVVNJc0ltTnlkaUk2SWxBdE1qVTJJaXdpYTNSNUlqb2lSVU1pZlN3
aWEybGtJam9pY0RKdFFraHRSRVZPYjBsVmNEWnZOMkpaUm5oMmVYTkJRMWR4VUZVeGRIUndkbTVoZFdsb1ZYVn
lSU0lzSW1WdVl5STZJa0V5TlRaSFEwMGlMQ0poYkdjaU9pSkZRMFJJTFVWVEluMC4ueXlHNzZzYVFMcVIwQU5L
Ry5PLU5vdm4wX3hLeVBtSnRaMHNJREFQYWlEb0R1cDB5dExRU2dLOG1vRTk0REloOVhBOWdKdjNwUXl6c3p0Rk
ZYY2tIWEExUTR3cUpNRVUzem55LXdDalpaRTM4Q2U3YWg3SEZhNnQxai1fdnpYUEM2cmpfNFRKZ1oyeUc0MlFn
REdwclBoMlV1MGJhRDNqQW1EU1lxZk9IV0V1X2VSV0U3RFhJZGt1MHZHTEdGc25BUzZ1SnM3TTluY2dVdF9MQT
VRNHljQ0s4NzlIWnhvMG5vbGlaTHVGcXBzRnA3OVZfWEZncmM2MEhLWC1YOFA2T0xrdHdvQkR1WjR4TTRIWk1k
dWVuNFp6azFPY1BXYUUxREV6WjE2c3dkVHZsZTFaSGJjTmJEaHpPRTdPVkx1cUppTjFWOHRjTDQzTkRNN3Vnel
BzMDdJcHhBYTYwcElBMGtGX2JRcmhzeGVtdU5hRkg5N1c0VExDZkZYMGhKR0twbjN3ajNDYW9OVWtpak9QaFNs
NVAzTjBic2RnVGxTNndZT1hObmp2Zk5DZTlXRk9lOGFJa192SDh4X0I1VlNoWjh6emxiSlRvZDNCTzVvb0RTVV
hOZno2WWZaMzJNbzJua19OVHpWcEg5c3Z2TVBhc0FkRzNKM1Z5eUg4UFgzckNuVkdwQVFvR3lsdmRzUk5McU5L
MDYzRjJnMWpkdjBzVHpaZy5fQWFXNXRMWDZjNFVTN25UR0lfSk93IiwiY29udGFpbmVyX2lkIjoiY0dGNWJXVn
VkRjlqYjI1MFlXbHVaQVhJNllqbGtORFF4T0RJdFl6TXhNaTAwT0RNNUxXRXhZMlV0TVdNMU5HVTVOV0kyWkFH
RmlYelV4TlRBME5UWXpOVGd6TkRFMU5qRVpEIiwibW9kZSI6IlRFU1QifQ.3k6coUCPnCISJp0Khjtf4axAOwJ
G-6N87AcIhB4rq40zI1-jBy_LDAInO1p9I4d6RpN8sn4yOjAF9T-Vs2LFug"
},
"orderId" : "MP090222",
"bin" : "000001",
"comments" : "Comments",
"walletType" : "4",
"paymentContext" : "20a22635-332d-4158-bd9c-d93a9af7fc45"
}
```


Response sample - Authorization and Authorization and Capture – Meta Checkout


```
{
    "ResponseCode": {
        "authorizationCode": "tst379",
        "mcRecurringAdvCode": "",
        "visaVbVRespCode": "",
        "procStatus": "0",
        "procStatusMessage": null,
        "approvalStatus": "1",
        "respCode": "00",
        "respCodeMessage": "Approved",
        "hostResponseCode": "100",
        "hostResponseCodeMessage": null,
        "avsRespCode": "B ",
```

```
        "hostAvsRespCode": "I3",
        "profileProcStatus": "",
        "profileProcStatusMessage": ""
    },
    "EnhancedAuth": {
        "ctiAffluentCard": null,
        "ctiCommercialCard": null,
        "ctiDurbinExemption": null,
        "ctiHealthcareCard": null,
        "ctiLevel3Eligible": null,
        "ctiPayrollCard": null,
        "ctiPrepaidCard": null,
        "ctiPINlessDebitCard": null,
        "ctiSignatureDebitCard": null,
        "ctiIssuingCountry": null,
        "ctiPrepaidReloadableCard": null
    },
    "txRefNum": "6319EEBB4CB74214E06BE01C1480C7D98CB85390",
    "remainingBalance": null,
    "requestedAmount": null,
    "redeemedAmount": null,
    "partialAuthOccurred": null,
    "hash1": "Test Hash",
    "hash2": "Test Hash",
    "lastFourFPAN": "3877",
    "lastFourDPAN": null,
    "orderId": "MP090222",
    "cardBrand": "VI",
    "customerRefNum": "",
    "customerName": "",
    "mitReceivedTransactionID": null
}
```

Debundle Only

Request elements – Debundle Only

The following table lists the elements for Debundle Only requests.

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
bin	N/A	**Bank Identification Number (BIN)** Transaction routing definition assigned by Merchant Services. 000001 = Stratus 000002 = Tandem (i.e., PNS)	M	6	N
encryptedPaymentBundle	N/A	**Encrypted Payment Bundle** The content should be in JSON format, as received from Apple or Google.	M	Var	A

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
publicKey	N/A	**Public Key** Decrypts the incoming bundle using the public key. • Required for Google Pay Debundle Only requests (for direct merchants). • This is the public key registered by the merchant on the Google Developer portal.	O	Var	AN
latitudeLongitude	N/A	**Latitude and Longitude** The latitude and longitude of the device making the request. Although it is a required data element, sending a specific geo-location is optional. Populate with `1,1` if not participating in geo-location. Format: ±180.99999,±180.99999	M	21	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
walletType	N/A	**Wallet Type** Wallet type used for making the request. Valid values: • 1 = Apple Pay • 2 = Google Pay • 4 = Reserved	M	1	N

Response elements – Apple Pay Debundle Only

The table below lists the elements for Debundle only responses.

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
ProcStatus	N/A	**Process Status** Identifies whether or not transactions are successful. 0 = Success All other values constitute an error condition. Refer to Response Handling for a list of errors.	M	6	AN
TokenData	N/A	**Token Data** The high-level parent element for the Apple Pay encrypted payment bundle.	M	Var	A
applicationPrimaryAccountNumber	TokenData	**Application Primary Account Number** Consumer Digital Payment Token (CDPT)	M	19	N
applicationExpirationDate	TokenData	**Application Expiration Date** Token expiration date <YYMMDD>	M	6	N

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
currencyCode	TokenData	**Currency Code** ISO 4217 currency code	M	3	AN
transactionAmount	TokenData	**Transaction Amount**	M	12	N
cardholderName	TokenData	**Cardholder Name**	O	Var	AN
deviceManufacturerIdentifier	TokenData	**Device Manufacturer Identifier** Hex encoded device manufacturer identifier	M	Var	AN
paymentDataType	TokenData	**Payment Data Type** Always 3-D Secure for in-app payments	M	8	AN
paymentData	TokenData	**Payment Data parent element** Parent element for payment data	M	N/A	N/A
onlinePaymentCryptogram	paymentData	**Online Payment Cryptogram** Cryptogram value associated with CDPT.	M	Var	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
eciIndicator	paymentData	**ECI Indicator** ECI indicator value associated with CDPT. • MC: null • AX: null • VI: 7 (or 5) • DI: 5	O	2	N

M = Mandatory, C = Conditional, O = Optional, N/A = Not Applicable

Response elements – Google Pay Debundle Only

The following table lists the elements for Google Pay Debundle Only responses.

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
ProcStatus	N/A	**Process Status** Identifies whether or not transactions are successful 0 = Success All other values constitute an error condition. Refer to Response Handling for a list of errors.	M	6	AN
TokenData	N/A	**TokenData** High-level parent element for the Apple Pay encrypted payment bundle.	M	Var	A
dpan	TokenData	**DPAN** Consumer Digital Payment Token (CDPT).	C	19	N
pan	TokenData	**PAN** Primary Account Number.	C	19	N

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
expirationYear	TokenData	**Expiration Year** Four-digit expiration year of the card (e.g., 2020).	M	4	N
expirationMonth	TokenData	**Expiration Month** Expiration month of the card, where 1 is January, 2 is February, etc.	M	2	N
gatewayMerchantId	TokenData	**Gateway Merchant ID** Chain account number assigned by Merchant Services. This account number will match that of the merchant host platform: • BIN 000001: 6-digit Stratus division number • BIN 000002: 12-digit Tandem (i.e., PNS) MID	M	12	N
messageExpiration	TokenData	**Message Expiration** Expiration date of the encrypted bundle in time stamp format (milliseconds).	M	13	N

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
paymentMethod	TokenData	**Payment Method** Valid values: • TOKENIZED_CARD:for DPAN OR Tokenized cards • CARD: for FPAN or Clear Cards	M	15	AN
messageId	TokenData	**Message ID** Randomly generated ID (by Google) for each payload.	M	Var	AN
authMethod	TokenData	**Authentication Method** Authentication method of the card transaction. Valid values: • 3DS: 3DSecure • CARD	M	4	AN
threedsCryptogram	TokenData	**3DS Cryptogram** 3D Secure cryptogram.	C	Var	AN

Element Name	Parent Element Name	Description	Required	Max Char	Field Type
ecIndicator	paymentData	**ECI Indicator** ECI indicator value associated with CDPT. • MC: null • AX: null • VI: 07 (or 05) • DI: 05	O	2	N

Debundle Only samples

Request sample – Debundle Only – Google Pay (DPAN)


```
{
      "bin": "000001",
    "encryptedPaymentBundle":{
"signature":
"MEQCIBwjMNv72+6WQK0DbRqzn8JXdBuuLoHHAVvhF/u6MpVNAiBD4eU4ZJm8HiECGyNQh4EdlZY9LuCIIbQqe
c63PfjIRw\u003d\u003d",
"protocolVersion":" ECv2",
"signedMessage":
"{\"encryptedMessage\":\"WIoPQ0GMl3SVpFxzojpSQ2GdQM2EUozLuS2Au2SmiB7XyDyGZajqCvSD7gWGF
EtywTYUDLIMZaoKr/vw1wggyH2uulg7kcOPhcQg4pFbUHwfkKY0DfCy5zWcyqFH2xy87ZlBbCpgPP4o215KVtD
sf/0hHOTOhcmqUM/V/WjgSJ28jckumhiWiJBPnlqaVVUV3/ipBBbb44dghe4D5XFCXDL31efmXDNaprUT9Ziy1
htZdPIim89rH+8HBd1gRNNafZdHYAUWNxvpnOOrj2AwlVDn3aR/3nIPQHrlFv6fS0ft3/Hc7nu87hFvXCgVgC9
0okgKBUYQSFg81UhMo923Q3i0qReWLeimZanpjti1Tncw7NXk9nSwqsRgEgpp7Bn/N6Bj/Abdfu/ww3jjjwJxk
u/bNyhSlAFTFS8S6MV1FeZmf0ZyBIZ0Fwy+deRjqkivzv5uTzQ76CKNdhNtM7nPaaNN6R1JzJrdduHQ1Yxp/Dp
uawvIO4fWomu8LjcM5N7G3RQ/+EPIUuzQe9ruvplhhWjvX1ssEAjEzVhSXoTY9WSsL89GItA2rA\\u003d\\u0
03d\",
\"ephemeralPublicKey\":\"BC4WWm7AfnWdaihEU3TaczA0oEEeCgzE0p7PwIpyymZxn1LvcTRjLP7OkDQds
eixUapHXSafTSgrqSNQtRnaZxs\\u003d\",
\"tag\":\"bnoj8gvxEEVsvpnA936PCTYAOKR/6FHukTDDg2CYWj0\\u003d\"}
"},
      "latitudeLongitude": "1,1",
       "publicKey":"BLP3PV/c0kkqiOYk8Zf96nNXs9YREjEpPz/6PfEX8Gpd1LKbCf31QziTgWj1bMEXvc
wqfS6MEPZ/k2jn9U9D81s=",
      "walletType": "2"
}
```


Response sample – Debundle Only – Google Pay (DPAN)


```
{
"ProcStatus": "0",
"TokenData": {
    "dpan": "520424******7840",
    "pan": null,
    "expirationYear": "2024",
    "expirationMonth": "12",
    "gatewayMerchantId": "041756",
    "messageExpiration": "1576699360187",
    "paymentMethod": "TOKENIZED_CARD",

"messageId":"AH2Ejtcs28szciAR4eja6r4nvGLbqHqE75E9C6pc5bRElIyY4SWhloZbqKG5xNNZcspRv7GBi
zBJX2iRV2O0eZkWfZ9vKjxrb7K5j8qsVeiUWp89jU1Lw8TxVieIQOXfSbwb609A7xDN",
    "authMethod":"3DS",
    "threedsCryptogram":"ALnt+yWSJdXBACMLLWMNGgADFA==",
    "eciIndicator":null
}
```


Request sample – Debundle Only – Google Pay (FPAN)


```
{
      "bin": "000001",
    "encryptedPaymentBundle":{
"signature":"MEUCIQDD+yp+wOEevwQAidf20/gz9PX3snzhT4ukVBBTy/kQoAIgKvMM1+MmVvzl6DTLqIyKf
B3n/fkmbZFFV0JLMzK/7xE\u003d",
"protocolVersion":" ECv2",
```

```
"signedMessage":"{
\"encryptedMessage\":\"VmGWuVlftbSjRYEjg00R0heqkqgVdYnPGqp/ghJPwENhGPQ3y0M8+v6vaxJXv3P
BAQ4wbpOBgLkYMdVKfHLX79fmSwldxAVaEO7mvIgydaovmEemactEh/uh5qasljVwNoSaqnUBJmVKI87vaUdsh
/3QSDBMkgNOqrzheKfiTbRTypO5AMufQYZR1wjMo4owqf/NXyPfb2o2UImJNreAHimAb/8RnV+BAU9/SKjU6ey
06SD3YuFFupHeQWnASRXJNgLYDUJyz/RGODPwHaz2627ZAY5p1PJ+3/nl03lrb8z/5PoWe20TMfE8076XdnQKY
buTec6TLhLIWzLUq6M1JchQrDK3uaaLowBDE50kO9hACn34fg214GV6DDxSVK4hjRo6TmwH8++ngfHmcDBro2e
+yEykB/9L58fX91Z64DZsQMYU66oMAgD7/TSbgpqFUQ\\u003d\\u003d\",
\"ephemeralPublicKey\":
\"BCy1IHNjmYaH3Ud50ddul/fUNNApfy+/n6kVHof60fC1y8P1gpr32a59z6zuiQhQd5na5ejT0a+vlrTFXLpp
/EE\\u003d\",
\"tag\":\"QPVyDE8M61Et7wqxPrzjQxdufQuz4wOEkcwR2IxX13o\\u003d\"}
"},
      "latitudeLongitude": "1,1",
      "walletType": "2"
}
```


Response sample – Debundle Only – Google Pay (FPAN)


```
{
    "ProcStatus": "0",
    "TokenData": {
    "dpan": null,
    "pan": "411111******1111",
    "expirationYear": "2024",
    "expirationMonth": "12",
    "gatewayMerchantId": "041756",
    "messageExpiration": "1576184097691",
    "paymentMethod": "CARD",
    "messageId":
"AH2EjtcbEgmUGRJLHutMTCVnH8uZCfTd6yG5UL6euHzp4paz1nPdCUjTLMwPQNPW3xdZxub7RLckmrYW1Skma
eiBv-6ObdyVlOznpOqcZzpD5YLTaHvuznvy0U5QSLLIkEPNiVw_lDHG",
"eciIndicator": null
}
```


Request samples – Debundle Only – Apple Pay


```
{
"bin": "000002",
"encryptedPaymentBundle" : { "data" :
"IzxSm6YWehmlLvk5HY/rsl4hhWuorOG7R6ERP0fqzTokMhS5JtyAU8ajPIu/aHcbOxYQOhvk/K+3n6N7SbEKg
SuT100YFmeIKh3IkSLa4u1/1Y4Z9y5bqZFPxd8IcQnuR8HZKgJDHCXQzDDYP4JBMtqZQzRztzsIfa4eoOnGuZC
c2s+WxGap4iv92vPj8tAHonvSE9t0ByUCBLgfvu25GR0eJb6UM8nBvxP2/qBSElOuyLo80enrZ6tlp3xtpBEV8
oeOc9iLSmalayfD7JQxZXd2cWA/sZPWn4VGIj7Dt05NYE/iFZrw2VOa2hOJ4/4dOGS1KJzhw+RPRufhadAF96k
7O3LwbMphcM9sZLN/Y/LSqVFGzIq6ZlrnOwcxzvjNqw4ccNl4v3eehL4TRRgfF3LirV56BeADzJmq0pB3W/vu"
,
"header" : {
"ephemeralPublicKey" :
"MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEQ3CCwyRLUK61yxYifPLYY87iWcPydTCL0PpAOkpOAvZDCCffK
bQTsxK9707qmVrAmH0wDNZEbLJ9Ob3teiiCbA==",
"publicKeyHash" : "MUwkjyUBpyRiZTVMUrIzA6+SIrr9mV8nNct6YO0rGNg=",
"transactionId" : "ad9256898767x9618750998af3058af5b8ede5fb67bbdc37370b289743f762a7"
},
"signature" :
"MIAGCSqGSIb3DQEHAqCAMIACAQExDzANBglghkgBZQMEAgEFADCABgkqhkiG9w0BBwEAAKCAMIIBYjCCAQigA
wIBAgIGAV1lOPsBMAoGCCqGSM49BAMDMDgxITAfBgNVBAMMGFBheW1lbnRlY2ggTW9iaWxlIFNESyBDQTETMBE
GA1UECgwKUGF5bWVudGVjaDAeFw0xNzA3MjExMjU2NTlaFw0zNDA3MjExMjU2NTlaMDgxITAfBgNVBAMMGFBhe
W1lbnRlY2ggTW9iaWxlIFNESyBDQTETMBEGA1UECgwKUGF5bWVudGVjaDBZMBMGByqGSM49AgEGCCqGSM49AwE
HA0IABEKuXMH9Q3bZlekeTuImojxPuHQnxA4jIKiFwF3wOH6nQY94asOmLLLws3JD9tv2M2P7ppUl961r15aw4
8Gnr2UwCgYIKoZIzj0EAwMDSAAwRQIhAIvcLMW83wgdvH0Mhi1ZJa93CV5bY6Ru5GKY/0vNb1F4AiBO4bPOqW7
YR8GlJ6x823vx+AATTg5gocYGrj8tquPnjQAAMYIBGzCCARcCAQEwQjA4MSEwHwYDVQQDDBhQYXltZW50ZWNoI
E1vYmlsZSBTREsgQ0ExEzARBgNVBAoMClBheW1lbnRlY2gCBgFdZTj7ATANBglghkgBZQMEAgEFAKBpMBgGCSq
```

GSIb3DQEJAzELBgkqhkiG9w0BBwEwHAYJKoZIhvcNAQkFMQ8XDTE3MDkxODEyMzA1NVowLwYJKoZIhvcNAQkEM
SIEIFOTICKavR26ewV/9jepdbFWNoASpvLan5brcCutlZHzMAoGCCqGSM49BAMCBEgwRgIhAO8S85/SS1fX0TR
yDu7RA5wO/lRTF2ayk1PPcE9IN7i3AiEApAP4zETvW3jpipxp/nrKcISIGSm+XTmHXCiJZB/vthMAAAAAAAA="
,
"version" : "EC_v1"
},
"latitudeLongitude": "1,1",
"walletType": "1"
}

Response sample – Debundle Only – Apple Pay

{
"ProcStatus": "0",
"TokenData": {
"applicationPrimaryAccountNumber": "477777******9990",
"applicationExpirationDate": "170430",
"currencyCode": "840",
"transactionAmount": 1000,
"deviceManufacturerIdentifier": "040010030273",
"paymentDataType": "3DSecure",
"paymentData": {
"onlinePaymentCryptogram": "BwAQA5SFcAEAABNZGYVwEM04oio=",
"eciIndicator": -1
}

Mark for Capture

Refer to MIME Header for all API calls.

Request elements – Mark for Capture

The table below lists the elements for Mark for Capture (MFC) requests.

Field	Description	Required	Max Char	Field Type
orderID	**Merchant-Defined Order Number** Must match the order ID of the original transaction being marked for capture.	M	22	AN
amount	**Amount to be captured keys**: • Implied decimal including those currencies that are a zero exponent. For example, both $100.00 (an exponent of 2) and ¥100 (an exponent of 0) should be sent as amount is 10000. • Amount can be less than, equal to or higher than the amount of the original transaction being marked for capture. If the amount submitted is less than the original transaction, the new order will be split. To submit an amount higher than the original transaction, contact your Technical Implementations Analyst to enable the feature. Refer to the Tandem Processing Integration Guide located on Developer Center for industry tolerances.	C	12	N

Field	Description	Required	Max Char	Field Type
taxInd	**Level 2 Data - Tax Type** The original transaction can be updated with level 2 data during an MFC request. Conditionally required if the level 2 data is being added via the MFC. • **0** = Not provided Included: • Non-taxable Note: Refer to **Level 2 and 3 Data Reference** in the **Orbital Gateway Web Service Interface Developer Guide** for further details.	C	1	N
taxAmount	**Level 2 Data - Tax Amount for the Purchase** The original transaction can be updated with level 2 data during a MFC. Conditionally required if the level 2 data is being added via MFC. Implied decimal, including those currencies that are a zero exponent. Note: Refer to **Level 2 and 3 Data Reference** in the **Orbital Gateway Web Service Interface Developer Guide** for further details.	C	12	N

Field	Description	Required	Max Char	Field Type
bin	**Transaction Routing Definition** Assigned by Merchant Services. • 000001 = Stratus • 000002 = Tandem (i.e., PNS)	M	6	N
merchantID	**Gateway Merchant Account Number Assigned by Merchant Services** This account number will match that of the user's host platform: • BIN 000001: 6-digit Stratus division number • BIN 000002: 12-digit Tandem (i.e., PNS) MID	M	15	N
terminalID	**Merchant Terminal ID Assigned by Merchant Services**	M	3	N
txRefNum	**Gateway Transaction Reference Number** A unique value for each transaction, which is required in order to adjust any transaction in the Orbital Gateway, such as a MFC or V.	M	40	AN

Field	Description	Required	Max Char	Field Type
retryTrace	**Trace Number Used for Retry Logic** Client generated number to determine the uniqueness of a transaction by recognizing subsequent retries of the same request. Required for PINless debit transactions. Note: Refer to **Retry Logic** in the **Orbital Gateway Web Service Interface Developer Guide** for information on this field.	C	16	N
pCardOrderID	**Level 2 data - PO Number from Customer** The original transaction can be updated with level 2 data during an MFC request. Conditionally required if level 2 data is being added via MFC. Do not include any of the following characters: <>?;':"[]\{}|`=~!@#%^&*()_+ Note: Refer to **Level 2 and 3 Data Reference** in the **Orbital Gateway Web Service Interface Developer Guide** for further details.	C	17	A

Field	Description	Required	Max Char	Field Type
pCardDestZip	**Level 2 Data - Shipping Destination Zip Code for the Purchase** The original transaction can be updated with level 2 data during an MFC request. Conditionally required if the level 2 data is being added via the MFC. For zip code + 4, separate with a hyphen (-). Note: Refer to **Level 2 and 3 Data Reference** in the **Orbital Gateway Web Service Interface Developer Guide** for further details.	C	10	AN
pCardDestName	**AX Purchasing Card Data – Cardholder Ship To: Name** The original transaction can be updated with purchasing card information during an MFC request. Stratus only/conditionally required if the AX purchasing card data is being added via MFC. Note: Refer to **Level 2 and 3 Data Reference** in the **Orbital Gateway Web Service Interface Developer Guide** for further details.	O	30	AN

Field	Description	Required	Max Char	Field Type
pCardDestAddress	**AX Purchasing Card Data - Cardholder Ship To: Address Line 1** The original transaction can be updated with purchasing card information during an MFC request. Stratus only/conditionally required if the AX purchasing card data is being added via MFC. Note: Refer to **Level 2 and 3 Data Reference** in the **Orbital Gateway Web Service Interface Developer Guide** for further details.	C	30	AN
pCardDestAddress2	**AX Purchasing Card Data - Cardholder Ship To: Address Line 2** The original transaction can be updated with the purchasing card information during an MFC request. Stratus only: Conditionally required if the AX purchasing card data is being added via an MFC. Note: Refer to **Level 2 and 3 Data Reference** in the **Orbital Gateway Web Service Interface Developer Guide** for further details.	O	30	AN

Field	Description	Required	Max Char	Field Type
pCardDestCity	**AX Purchasing Card Data – Cardholder Ship To: City** The original transaction can be updated with purchasing card information during an MFC request. Stratus only: Conditionally AX purchasing card data. Note: Refer to **Level 2 and 3 Data Reference** in the **Orbital Gateway Web Service Interface Developer Guide** for further details.	C	20	AN
pCardDestStateCd	**AX Purchasing Card Data – Cardholder Ship To: State** The original transaction can be updated with purchasing card information during an MFC request. Stratus only: Conditionally for AX purchasing card data. Note: Refer to **Level 2 and 3 Data Reference** in the **Orbital Gateway Web Service Interface Developer Guide** for further details.	C	2	AN

Field	Description	Required	Max Char	Field Type
amexTranAdvAddn1	**AX Purchasing Card Data - Transaction Advice Addendum 1** The Transaction Advice Addenda (TAA) record is used to further identify the purchase associated with a charge to the cardholder. The TAA is also used in purchasing/procurement card transactions to provide specific details regarding the transaction to the cardholder for tracking purposes. TAAs should be as concise as possible, while still providing adequate information. For example, a TAA of merchandise would not be acceptable. Stratus only: Required for AX purchasing card data. Note: Refer to **Level 2 and 3 Data Reference** in the **Orbital Gateway Web Service Interface Developer Guide** for further details.	C	40	AN
amexTranAdvAddn2	**AX Purchasing Card Data - Transaction Advice Addendum 2** The original transaction can be updated with purchasing card information during an MFC request. Stratus only: Conditionally required for AX purchasing card data. Note: Refer to **Level 2 and 3 Data Reference** in the **Orbital Gateway Web Service Interface Developer Guide** for further details.	C	40	AN

Field	Description	Required	Max Char	Field Type
amexTranAdvAddn3	**AX Purchasing Card Data - Transaction Advice Addendum 3** The original transaction can be updated with purchasing card information during an MFC request. Stratus only: Conditionally required for AX purchasing card data. Note: Refer to **Level 2 and 3 Data Reference** in the **Orbital Gateway Web Service Interface Developer Guide** for further details.	C	40	AN
amexTranAdvAddn4	**AX Purchasing Card Data - Transaction Advice Addendum 4** The original transaction can be updated with purchasing card information during an MFC request. Stratus only: Conditionally required for AX purchasing card data. Note: Refer to **Level 2 and 3 Data Reference** in the **Orbital Gateway Web Service Interface Developer Guide** for further details.	C	40	AN

Field	Description	Required	Max Char	Field Type
digitalTokenCryptogram	**Digital Token Cryptogram** The gateway will parse any data provided in this field and convert it into the proper format. **For Discover CDPT transactions:** • This number must be base 64 encoded. • Cryptographic value derived with an algorithm that applies the issuer's private key to the combination of the cardholder account number, transaction identifier and other data. **For Consumer Digital Payment Tokens:** • Unique transaction cryptogram generated by the digital wallet provider, and should be submitted as it was received. • Note: Merchants must use this field to submit a new Accountholder Authentication Value (AAV)/CAVV when a new authorization is needed.	C	120	AN

Field	Description	Required	Max Char	Field Type
authenticationECIInd	**Transaction Type** Conditionally required for VI Secure, MC Identity Check, DI ProtectBuy and AX SafeKey transactions. • 2 = Designates a recurring transaction conducted with a CDPT • 5 = VI Secure, MC Identity Check, DI ProtectBuy or AX SafeKey – authenticated transaction or an e-commerce CDPT • 6 = VI Secure, MC Identity Check, DI ProtectBuy or AX SafeKey – attempted authentication • 7 = Secure e-commerce with SSL/TLS encryption - authentication not performed • 20 = Designates an AX CDPT Note: For CDPTs, this field may be left empty. The Orbital Gateway will derive the appropriate value.	C	2	N

Field	Description	Required	Max Char	Field Type
ucafInd	**Universal Cardholder Authentication Field (UCAF) Collection Indicator** Indicates merchant support of the UCAF. Valid values: • Blank = Merchant does not support, or opted not to send, UCAF • 1 = Attempted authentication data present • 2 = Full authentication data present • 3 = Static authentication data present • 4 = Not authenticated, data only call* • 5 = Issuer risk-based decisioning • 6 = Merchant Risk Based Decisioning (MDES) token • 7 = Partial shipment or recurring payment For MC transactions, if the UCAF is **4**, include `mcProgramProtocol` and `mcDirectoryTransID`. Supported Methods of Payment (MOPs) include MC, International Maestro (IM), and MR.	O	1	N

M = Mandatory, C = Conditional, O = Optional

Response elements – Mark for Capture

The table below lists elements for MFC responses.

Field	Description	Required	Max Char	Field Type
bin	Transaction routing definition echoes BIN sent in request.	M	6	N
merchantID	**Gateway merchant account number assigned by Merchant** Services. Echoes the MID sent in request.	M	15	N
terminalID	Merchant terminal ID assigned by Merchant Services echoes the terminal ID sent in request.	M	3	N
orderID	**Merchant-Defined Order Number** Field defined and supplied by the authorization originator, and echoed back in response.	C	22	AN
txRefNum	**Gateway Transaction Reference Number** A unique value for each transaction, which is required in order to adjust any transaction in the Gateway (e.g., MFC or V request).	M	40	AN

Field	Description	Required	Max Char	Field Type
txRefIdx	**Gateway Transaction Index** Used to identify the unique components of transactions adjusted more than one time. Required on V transactions.	M	4	AN
splitTxRefIdx	**Transaction Reference Number of Split Transaction** Returns the transaction reference number of the partial an MFC request.	C	40	AN
amount	**Amount captured**	C	12	N
respDateTime	**Date/time the transaction was processed by Gateway** Format: MMDDYYYYhhmmss	M	14	N

Field	Description	Required	Max Char	Field Type
procStatus	**Process Status** The first data set that should be checked to determine the result of a request. The only element that is returned in all response scenarios. Identifies whether transactions have successfully passed all Gateway edit checks: 0 = Success All other values constitute an error condition.	M	6	AN
procStatusMessage	Text message associated with `respCode` value.	C	Var	A
retryTrace	Defines the trace number used for retry logic echo of request value, if sent.	C	16	N
retryAttempCount	Number of times a transaction result has been returned • 0 = First response (unique `retryTrace`) • ≥1 = Orbital Gateway has processed this request previously and is echoing back the response. The number represents the number of requests processed by the Orbital Gateway with the same `retryTrace` number.	C	2	N

Field	Description	Required	Max Char	Field Type
lastRetryDate	**Date of Last Retry Attempt** The date/time at which the previous transaction using the same `retryTrace` value was processed by the Orbital Gateway. Format: yyyymmddhh24mmss	C	14	N

M = Mandatory, C = Conditional, O = Optional

Mark for Capture samples

Request sample – Mark for Capture

```
{
"orderID": "587469",
"amount": "100",
"taxInd": "1",
"taxAmount": "100",
"bin": "000001",
"merchantID": "041756",
"terminalID": "001",
"txRefNum": "5E6284AF52EDFD5866BB925FB51EB531140F53C8",
"retryTrace": "",
"pCardOrderID": "",
"pCardDestZip": "",
"pCardDestName": "",
"pCardDestAddress": "",
"pCardDestAddress2": "",
"pCardDestCity": "",
"pCardDestStateCd": "",
"amexTranAdvAddn1": "",
"amexTranAdvAddn2": "",
"amexTranAdvAddn3": "",
"amexTranAdvAddn4": "",
"digitalTokenCryptogram": "",
"authenticationECIInd": "",
"ucafInd": ""
}
```

Response sample – Mark for Capture

```
{
    "bin": "000001",
    "merchantID": "041756",
    "terminalID": "001",
    "orderID": "587469",
    "txRefNum": "5E6284AF52EDFD5866BB925FB51EB531140F53C8",
    "txRefIdx": "3",
    "splitTxRefIdx": "5E6286B7EAAE62C85886C461C9F610E3F3AF5369",
    "amount": "100",
    "respDateTime": "20200306122159",
    "procStatus": "0",
    "procStatusMessage": "Approved",
    "retryTrace": "",
    "retryAttempCount": "",
    "lastRetryDate": ""
}
```

Profile Management

There are Uniform Resource Locators (URLs) associated with each profile management action. The definitions associated with the URLs are listed below. The profile actions are as follows:

- Create (Add)
- Update
- Fetch (Retrieve)

Refer to MIME Header for all API calls.

Create (Add) Profile

Request elements – Create Profile

The table below lists the elements to create a profile request.

Element Name	Description	Required	Max Char	Field Type
bin	**Transaction Routing Definition** Assigned by Merchant Services. Valid values: • 000001 = Stratus • 000002 = Tandem (i.e., PNS)	M	6	N
merchantID	**Gateway Merchant Account Number Assigned by Merchant Services** This account number will match that of the host platform: • BIN 000001: 6-digit Stratus division number • BIN 000002: 12-digit Tandem (i.e., PNS) MID	M	15	N
customerName	**Customer Billing Name** Conditionally required for electronic check profiles. This is the equivalent to the `avsName` element used during transactional requests.	C	30	AN

Element Name	Description	Required	Max Char	Field Type
customerRefNum	Sets the customer reference number used to apply a customer profile on all future orders. Required if: • Creating a profile **and** the `customerProfileFromOrderInd` option is **S** (use the `customerRefNum` field). • If `customerProfileFromOrderInd` is **A**, the customer reference number will be defined by the Orbital Gateway, and any value passed in this element will be ignored. Given this value can be the same as the order number, the valid characters for this field follow the same conventions as the order ID element, and include: • Alphabetical (lower- and upper-cased) • Numeric (0 – 9) • - , $, @, &, and a space character (space cannot be the leading character) Note: All alphabetic characters in this field are stored in uppercase by the Orbital Gateway. Upper- and lower-cased values cannot be used to differentiate customer reference numbers. This value cannot be changed through a profile update action.	C	22	AN

Element Name	Description	Required	Max Char	Field Type
customerAddress1	**Cardholder Billing Address Line 1** Optional if creating a profile. This is the equivalent to the `avsAddress1` element used during transactional requests.	O	30	AN
customerAddress2	**Cardholder Billing Address Line 2** Optional if creating a profile. This is the equivalent to the `avsAddress2` element used during transactional requests.	O	30	AN
customerCity	**Cardholder Billing City** Optional if creating a profile. This is the equivalent to the `avsCity` element used during transactional requests.	O	20	AN
customerState	**Cardholder Billing State** Optional if creating a profile. This is the equivalent to the `avsState` element used during transactional requests.	O	2	AN

Element Name	Description	Required	Max Char	Field Type
customerZIP	**Cardholder Billing Address Zip Code** Equivalent to the `avsZip` element used during transactional requests. Required for all AVS requests. Must include the 5-digit zip code at a minimum. Separate zip code + 4 with a hyphen (-). Note: To avoid declined transactions, always send full Address Verification Services (AVS) data.	O	10	AN
customerEmail	**Cardholder Email Address** Optional if creating a profile.	O	50	AN

Element Name	Description	Required	Max Char	Field Type
customerPhone	**Cardholder Telephone Number** AAAEEENNNNXXXX, where: • AAA = Area code • EEE = Exchange • NNNN = Number • XXXX = Extension Optional if creating a profile. This is the equivalent to the `avsPhone` element used during transactional requests.	O	14	AN
customerCountryCode	**Cardholder Billing Address Country Code** Valid values: • US = United States • CA = Canada • GB = Great Britain • UK = United Kingdom This is the equivalent to the avsC`ountryCode` element used during transactional requests.	C	2	A

Element Name	Description	Required	Max Char	Field Type
customerProfileOrderOverideInd	Defines if any order data can be pre-populated from the customer reference number (`customerrefnum`) Required if creating a profile. Valid values: • NO = No mapping to order data • OI = Use `customerrefnum` for `orderID` • OD = Use `customerrefnum` for `comments` • OA = Use `customerrefnum` for `orderID` and `comments`	C	2	A
customerProfileFromOrderInd	**Customer Profile Number Generation Options** Required if creating a profile, as well as defines the customer profile number. • A = Auto-generate the `customerRefNum` • S = Use the `customerRefNum` element	C	5	A

Element Name	Description	Required	Max Char	Field Type
orderDefaultDescription	**Order Description** Optional if creating a profile. If `customerProfileOrderOverrideInd` = `OA`, do not set this value, since this defaults the order description to the customer reference number. This is the equivalent to the c`omments` element used during transactional requests.	O	64	A
orderDefaultAmount	**Transaction Amount** Optional if creating a profile. This is the equivalent to the `<Amount>` element used during transactional requests. Keys: • Implied decimal, including those currencies that are a zero exponent. For example, both $100.00 (an exponent of 2) and ¥100 (an exponent of 0) should be sent as `<orderDefaultAmount>10000</orderDefaultAmount>` • Given that each Orbital Gateway MID is restricted to one currency, the currency code (and exponent) is defaulted based on the MID in which a transaction is presented	O	12	N

Element Name	Description	Required	Max Char	Field Type
customerAccountType	**Customer's payment type to save in the profile**. Required if creating a profile. Valid values: • **CC** = Credit card • **CR** = ChaseNet signature debit • **CZ** = ChaseNet credit card	M	2	A
ccAccountNum	**Customer Credit Card Number** This is the equivalent to the `ccAccountNum` element used during transactional requests.	C	19	AN

Element Name	Description	Required	Max Char	Field Type
ccExp	**Customer Credit Card Expiration Date** Required when creating a profile and `customerAccountType` is **CC** Format: MMYY or YYYYMM Stratus (BIN 000001) allows a **blank** to be submitted when no known expiration date exists. There are three valid mechanisms for submitting a **blank** expiration date to the Stratus host using Orbital: • Null-fill the element • Send four spaces • Zero-fill Note: Discuss this feature with a Merchant Services Integration Testing Analyst before implementing.	M	6	N

Element Name	Description	Required	Max Char	Field Type
mbType	**Managed Billing Type** Indicates the type of managed billing in which the merchant is participating. • R = Recurring • D = Deferred The value submitted must be in agreement with the type of managed billing for which the merchant is configured by Merchant Services. This field serves to notify the Orbital Gateway that the transaction is a managed billing transaction. If this field is not sent with a managed billing transaction, all other managed billing fields are ignored.	C	1	A
mbOrderIdGenerationMethod	**Managed Billing Order ID Generation Method** Required for managed billing transactions, and sets the method by which Orbital uses to generate the order ID. Note: This field does **not** influence the order ID for stand-alone transactions initiated by the merchant. Valid values: • IO = Use the customer reference number (profile ID) • DI = Dynamically generate the order ID	C	2	A

Element Name	Description	Required	Max Char	Field Type
mbRecurringStartDate	**Managed Billing Recurring Start Date** Defines the future date at which Orbital begins a recurring billing cycle with the associated profile. Required when `mbType` is not null. Must be at least one day following the request date in order to allow the managed billing engine to properly calculate and schedule payments. A recurring billing cycle cannot begin on the date at which the request message is sent to the Orbital Gateway. Format: MMDDYYYY	C	8	N
mbRecurringEndDate	**Managed Billing Recurring End Date** Defines the future date at which Orbital ends a recurring billing cycle with the associated profile. Format: MMDDYYYY This is the first of three possible recurring end triggers. Only one end trigger can be submitted per request message.	C	8	N

Element Name	Description	Required	Max Char	Field Type
mbRecurringNoEndDateFlag	**Managed Billing "No End Date" Indicator** Valid values: • Y = Schedules recurring transactions for an infinite amount of time. A Y in this element overrides the value, if any, in the `mbRecurringEndDate` field. • N (or blank) = Orbital uses the value of the `mbRecurringEndDate` field to define the recurring end date. This is the second of three possible recurring end triggers. Only one end trigger can be submitted per request message.	C	1	A
mbRecurringMaxBillings	**Managed Billing Maximum Number of Billings** This value defines the maximum number of billings permitted for a recurring billing cycle. Valid values: 1–999999 This is the third of three possible recurring end triggers. Only one end trigger can be submitted per request message.	C	6	N

Element Name	Description	Required	Max Char	Field Type
mbRecurringFrequency	**Managed Billing Recurring Frequency Pattern** This pattern is a subset of a standard CRON expression, comprising 3 fields separated by white space: Fields: • Day of month • Month • Day of week Permitted values: • 1-31 (for day of month) • 1-12 (for January - December) • 1-7 (for day of week) Permitted special characters: • , - * ? / L W (for day of month) • , - * / (for month) • , - * ? / L # (for day of week)	C	64	AN

Element Name	Description	Required	Max Char	Field Type
mbDeferredBillDate	**Managed Billing Deferred Billing Date** Defines the future date at which Orbital triggers a one-time billing with the associated profile. This date must be at least one day following the request date. A deferred billing cannot take place on the date at which the request message is sent to the Orbital Gateway. Format: MMDDYYYY	C	8	N

Element Name	Description	Required	Max Char	Field Type
softDescMercName	**Soft Descriptor Merchant Name** Conditionally required for soft descriptors. The **Merchant Name** field should be the most recognizable to the cardholder (e.g., company or trade name). The actual length of this field is conditionally tied to the host, as well as the size of the `softDescProdDesc` element used. **Stratus:** Credit – Three options that conditionally affect the `softDescProdDesc` are as follows: • Maximum 3 bytes • Maximum 7 bytes • Maximum 12 bytes **Tandem (i.e., PNS):** • Maximum 25 bytes	C	25	A

Element Name	Description	Required	Max Char	Field Type
softDescProdDesc	**Soft Descriptor Product Description** Conditionally required for soft descriptors. Provides an accurate description. Stratus: Credit: • If `softDescMercName` = 3 bytes (maximum of 18) • If `softDescMercName` = 7 bytes (maximum of 14) • If `softDescMercName` = 12 bytes (maximum of 9) Tandem (i.e., PNS): • This field does not display on cardholder statements for Tandem/PNS merchants.	C	18	A
softDescMercCity	**Soft Descriptor Merchant City** Merchant city for retail. Required for soft descriptors. Any soft descriptor element that is not populated should be null-filled.	C	13	AN

Element Name	Description	Required	Max Char	Field Type
softDescMercPhone	**Soft Descriptor Merchant Phone** Tag required for soft descriptors. Only one location soft descriptor value should be sent (e.g., phone, URL, or email). All others should be null-filled. This field does not display on cardholder statements for Tandem (i.e., PNS) merchants. Valid formats: • NNN-NNN-NNNN • NNN-AAAAAAA Note: For BIN 000001 merchants processing MC (MOTO and recurring), if the **City/Phone** field at the division level is not a customer service phone number, then one must be populated or the transaction will be rejected, with a response reason code **BP** (i.e., missing customer service phone).	C	12	AN
softDescMercURL	**Soft Descriptor Merchant URL** Tag conditionally required for soft descriptors. Only one location soft descriptor value should be sent (e.g., phone, URL, or email). All others should be null-filled. This field does not display on cardholder statements for Tandem (i.e., PNS) merchants.	C	13	AN

Element Name	Description	Required	Max Char	Field Type
softDescMercEmail	**Soft Descriptor Merchant Email** Tag conditionally required for soft descriptors. Only one location soft descriptor value should be sent (e.g., phone, URL, or email). All others should be null-filled. This field does not display on cardholder statements for Tandem (i.e., PNS) merchants.	C	13	AN
status	**Profile Status Flag** This field is used to set the status of a customer profile. Valid values: • A = Active • I = Inactive • MS = Manual Suspend	C	2	A

Element Name	Description	Required	Max Char	Field Type
dpanInd	**CDPT Indicator** Used to identify the type of CDPT transaction submitted within an authorization. Valid values: • Y = For initial authorization • S = For subsequent or recurring authorizations Note: Once **Y** is sent, response is **19758 Invalid DPAN indicator value. Valid values are [S]**.	C	1	A
mitMsgType	**CIT Code** Indicates the message type used for the message type records.	C	4	A
mitSubmittedTransactionID	**Submitted TXID** The submitted TXID returned to the merchant from a previous CIT transaction within a series of transactions.	C	15	AN

M = Mandatory, C = Conditional, O = Optional

Request sample – Create Profile


```
{
    "bin": "000001",
    "merchantID": "041756",
    "customerName": "New customer",
    "customerAddress1": "4200 W Cypress St",
    "customerAddress2": "ste 350",
    "customerCity": "Tampa",
    "customerState": "FL",
    "customerZIP": "33607",
    "customerEmail": "email@email.com",
    "customerPhone": "8001112222",
    "customerCountryCode": "US",
    "customerProfileOrderOverideInd": "NO",
    "customerProfileFromOrderInd": "A",
    "orderDefaultDescription": "test",
    "orderDefaultAmount": "1550",
    "customerAccountType": "CC",
    "ccAccountNum": "545454******5454",
    "ccExp": "202109",
    "status": "A",
    "dpanInd": "S",
    "mitMsgType": "CREC",
    "mitSubmittedTransactionID": "010041692161776"
}
```

Update Profile

Request elements – Update Profile

The table below lists the elements to update a profile request.

Element Name	Description	Required	Max Char	Field Type
bin	**Transaction Routing Definition** Assigned by Merchant Services. Valid values: • 000001 = Stratus • 000002 = Tandem (i.e., PNS)	M	6	N
merchantID	**Gateway Merchant Account Number Assigned by Merchant Services** This account number matches that of the host platform: • BIN 000001: 6-digit Stratus division number • BIN 000002: 12-digit Tandem (i.e., PNS) MID	M	15	N
customerName	**Customer Billing Name** Conditionally required for electronic check profiles. This is the equivalent to the `avsName` element used during transactional requests.	C	30	AN

Element Name	Description	Required	Max Char	Field Type
customerRefNum	**The Customer Reference Number to be Modified** This value cannot be changed through a profile update action.	M	22	AN
customerAddress1	**Cardholder Billing Address Line 1** This is the equivalent to the `avsAddress1` field used during new order requests.	O	30	AN
customerAddress2	**Cardholder Billing Address Line 2** This is the equivalent to the `avsAddress2` field used during new order requests.	O	30	AN
customerCity	**Cardholder Billing City** This is the equivalent to the `avsCity` field used during new order requests.	O	20	AN
customerState	**Cardholder Billing State** This is the equivalent to the `avsState` field used during transactional requests.	O	2	AN

Element Name	Description	Required	Max Char	Field Type
customerZIP	**Cardholder Billing Address Zip Code** Equivalent to the **AVSzip** element used during transactional requests. Required for all AVS requests. • Must include the 5-digit zip code at a minimum. • Separate zip code + 4 with a hyphen (-). Note: To avoid declined transactions, always send full Address Verification Services (AVS) data.	O	10	AN
customerEmail	**Cardholder Email Address** Optional for updating a profile. There is no equivalent field available on new order requests.	O	50	AN
customerPhone	**Cardholder Telephone Number** AAAEEENNNNXXXX, where: • **AAA** = Area Code • **EEE** = Exchange • **NNNN** = Number • **XXXX** = Extension Optional for profile update. This is the equivalent to the `avsPhone` element used during transactional requests.	O	14	AN

Element Name	Description	Required	Max Char	Field Type
customerCountryCode	**Cardholder Billing Address Country Code** Valid values: • **US** = United States • **CA** = Canada • **GB** = Great Britain • **UK** = United Kingdom This is the equivalent to the `avsCountryCode` element used during transactional requests.	C	2	A
customerProfileOrderOverrideInd	Defines whether order data can be pre-populated from the customer reference number (**CustomerRefNum**) Optional for profile update requests. Valid values: • **NO** = No mapping to order data • **OI** = Use `customerRefNum` for `orderID` • **OD** = Use `customerRefNum` for`comments` • **OA** = Use `customerRefNum` for `orderID` and `comments`	C	2	A

Element Name	Description	Required	Max Char	Field Type
orderDefaultDescription	**Order Description** Optional values submitted in this field set a default value for the **comments** field (used on new order requests) that use this profile. If `customerProfileOrderOverrideInd` = `OA`, do not set this value, as this defaults the order description to the customer reference number. This is the equivalent to the `comments` element used during transactional requests.	O	64	A
orderDefaultAmount	**Transaction Amount** This is the equivalent to the `amount` element used during transactional requests. Keys: Implied decimal, including those currencies that are a zero exponent. For example, both $100.00 (an exponent of 2) and ¥100 (an exponent of 0) should be sent as `<orderDefaultAmount>10000</orderDefaultAmount>`. Given that each Orbital gateway MID is restricted to one currency, the currency code (and exponent) is defaulted, based on the MID in which the transaction is presented.	O	12	N

Element Name	Description	Required	Max Char	Field Type
customerAccountType	**Customer's Payment Type to be Saved in the Profile** Required if the account type is being changed. Valid values: • CC = Credit card • CR = ChaseNet signature debit • CZ = ChaseNet credit card	C	2	A
ccAccountNum	**Customer Credit Card Number** Optional field for profile update request. This is the equivalent to the cc`AccountNum` element used during transactional requests.	O	19	AN

Element Name	Description	Required	Max Char	Field Type
ccExp	**Customer Credit Card Expiration Date** Optional if updating a profile **and** `customerAccountType` is `CC` Format: MMYY or YYYYMM Stratus (BIN 000001) allows a blank to be submitted in cases where no known expiration date exists. Three valid mechanisms are used for submitting a blank expiration date to the Stratus host using Orbital: • Null-fill the element • Send four spaces • Zero-fill Note: It is suggested to discuss this feature with an Integration Testing Analyst prior to implementation.	O	6	N

Element Name	Description	Required	Max Char	Field Type
mbType	**Managed Billing Type** Indicates the type of managed billing in which the merchant is participating. • R = Recurring • D = Deferred The value submitted must be in agreement with the type of managed billing for which the merchant is configured by Merchant Services. This field serves to notify the Orbital Gateway that the transaction is a managed billing transaction. If this field is not sent with a managed billing transaction, all other managed billing fields are ignored.	C	1	A
mbOrderIdGenerationMethod	**Managed Billing Order ID Generation Method** Required for managed billing transactions, and sets the method Orbital to generate an order ID. Note: This field does **not** influence the order ID for stand-alone transactions initiated by the merchant. Valid values: • IO = Use the customer reference number (profile ID). This value is made up of the capital letters I and O, not numbers. • DI = Dynamically generate the order ID. This value is made up of the capital letters D and I, and no numbers.	C	2	A

Element Name	Description	Required	Max Char	Field Type
mbRecurringStartDate	**Managed Billing Recurring Start Date** Defines the future date at which Orbital begins a recurring billing cycle with the associated profile. Required when `mbType` is not null. Must be at least one day following the request date in order to allow the managed billing engine to properly calculate and schedule payments. A recurring billing cycle can never begin on the date at which the request message is submitted to the Orbital Gateway. Format: MMDDYYYY	C	8	N
mbRecurringEndDate	**Managed Billing Recurring End Date** Defines the future date at which Orbital ends a recurring billing cycle with the associated profile. Format: MMDDYYYY This is the first of three possible recurring end triggers. Only one end trigger may be submitted per request message.	C	8	N

Element Name	Description	Required	Max Char	Field Type
mbRecurringNoEndDateFlag	**Managed Billing "No End Date" Indicator** Valid values: • Y = Schedules recurring transactions for an indefinite amount of time. A Y in this element overrides the value, if any, in the `mbRecurringEndDate` field. • N (or blank) = Orbital uses the value of the `mbRecurringEndDate` field to define the recurring end date. This is the second of three possible recurring end triggers. Only one end trigger can be submitted per request message.	C	1	A
mbRecurringMaxBillings	**Managed Billing Maximum Number Of Billings** This value defines the maximum number of billings permitted for a recurring billing cycle. Valid values: 1–999999 This is the third of three possible recurring end triggers. Only one end trigger may be submitted per request message.	C	6	N

Element Name	Description	Required	Max Char	Field Type
mbRecurringFrequency	**Managed Billing Recurring Frequency Pattern** This pattern is a subset of a standard CRON expression, comprising 3 fields separated by white space. Fields: • Day of month • Month • Day of week Permitted values: • 1-31 (for day of month) • 1-12 (for January - December) • 1-7 (for day of week) Permitted special characters: • , - * ? / L W (for day of month) • , - * / (for month) • , - * ? / L # (for day of week)	C	64	AN

Element Name	Description	Required	Max Char	Field Type
mbDeferredBillDate	**Managed Billing Deferred Billing Date** Defines the future date at which the Orbital Gateway triggers a one-time billing with the associated profile. This date must be at least one day following the request date. A deferred billing cannot take place on the date at which the request message is sent to the Orbital Gateway. Format: MMDDYYYY	C	8	N
mbCancelDate	**Managed Billing Cancel Date** This field is used to cancel a single future billing that is already scheduled. The exact date of the scheduled billing must be submitted. Format: MMDDYYYY	C	8	N
mbRestoreDate	**Managed Billing Restore Billing Date** This field is used to reinstate a cancelled billing. The exact date of the previously-scheduled billing must be submitted in order for this action to function. Format: MMDDYYYY	C	8	N

Element Name	Description	Required	Max Char	Field Type
mbRemoveFlag	**Managed Billing Remove Flag** Valid values: • Y = This value is used to remove all managed billing settings from the associated profile. The profile becomes a standard profile, and any scheduled future billings are removed from the Orbital Gateway. • N (or blank) = This value has no effect on the profile.	C	1	A
softDescMercName	**Soft Descriptor Merchant Name** Conditionally required for soft descriptors. The **Merchant Name** field should be the most recognizable to the cardholder (e.g., company or trade name). The actual length of this field is conditionally tied to the host, as well as the size of the `softDescProdDesc` element used. **Stratus:** Credit – Three options that conditionally affect the `softDescProdDesc` are as follows: • Maximum 3 bytes • Maximum 7 bytes • Maximum 12 bytes **Tandem (i.e., PNS):** • Maximum 25 bytes	C	25	A

Element Name	Description	Required	Max Char	Field Type
softDescProdDesc	**Soft Descriptor Product Description** Conditionally required for soft descriptors. Provides an accurate description. Stratus: Credit: • If `softDescMercName` = 3 bytes (maximum of 18 bytes) • If `softDescMercName` = 7 bytes (maximum of 14 bytes) • If `softDescMercName` = 12 bytes (maximum of 9 bytes) Tandem (i.e., PNS): • This field does not display on cardholder statements for Tandem/PNS merchants.	C	18	A
softDescMercCity	**Soft Descriptor Merchant City** Tag is conditionally required for soft descriptors. The **Merchant City for Retail** field is required, but should be null-filled if any soft descriptor data is submitted.	C	13	A

Element Name	Description	Required	Max Char	Field Type
softDescMercPhone	**Soft Descriptor Merchant Phone** Conditionally required for soft descriptors. Only one location soft descriptor value should be sent (e.g., phone, URL, or email). All others should be null-filled. This field does not display on cardholder statements for Tandem (i.e., PNS) merchants. Valid formats: • NNN-NNN-NNNN • NNN-AAAAAAA Note: For BIN 000001 merchants processing MC (MOTO and recurring), if the **City/Phone** field at the division level is not a customer service phone number, one must be populated, or the transaction will be rejected with a **BP** response reason code (i.e., missing customer service phone).	C	12	AN
softDescMercURL	**Soft Descriptor Merchant URL** Conditionally required for soft descriptors. Only one location soft descriptor value should be submitted (e.g., phone, URL, or email). All others must be null-filled. This field does not display on cardholder statements for Tandem (i.e., PNS) merchants.	C	13	AN

Element Name	Description	Required	Max Char	Field Type
softDescMercEmail	**Soft Descriptor Merchant Email** Conditionally required for soft descriptors. Only one location soft descriptor value should be submitted (e.g., phone, URL, or email). All others must be null-filled. This field will not display on cardholder statements for Tandem (i.e., PNS) merchants.	C	13	AN
status	**Profile Status Flag** This field is used to set the customer's profile status. Valid values: • A = Active • I = Inactive • MS = Manual Suspend	C	2	A

Element Name	Description	Required	Max Char	Field Type
dpanInd	**Consumer Digital Payment Token Indicator** Used to identify the type of CDPT transaction submitted during an authorization. Refer to the References section for additional documentation on CDPT. Valid values: • Y = For initial authorization • S = For subsequent or recurring authorizations Note: Once **Y** is sent, response is `19758 Invalid DPAN Indicator Value. Valid values are [S].`	C	1	A

Request sample – Update Profile


```
{
    "bin": "000001",
    "merchantID": "041756",
    "customerName": "New customer",
    "customerRefNum": "1115504102",
    "customerAddress1": "4200 W Cypress St",
    "customerAddress2": "ste 350",
    "customerCity": "Tampa",
    "customerState": "FL",
    "customerZIP": "33607",
    "customerEmail": "email@email.com",
    "customerPhone": "8001112222",
    "customerCountryCode": "US",
    "customerProfileOrderOverideInd": "NO",
    "customerProfileFromOrderInd": "A",
    "orderDefaultDescription": "test",
    "orderDefaultAmount": "1550",
    "customerAccountType": "CC",
   "ccAccountNum": "545454******5454",
    "ccExp": "201809",
    "status": "A"
}
```

Fetch (Retrieve) Profile

Request elements – Fetch (Retrieve) Profile

The table below lists the elements to fetch (retrieve) a profile request.

Element Name	Description	Required	Max Char	Field Type
bin	**Transaction Routing Definition** Assigned by Merchant Services. Valid values: • 000001 = Stratus • 000002 = Tandem (i.e., PNS)	M	6	N
customerName	**Customer Billing Name** This is the equivalent to the `avsName` element used during transactional requests.	O	30	AN
customerRefNum	**Customer Reference Number on profile to be Retrieved** This value cannot be changed through a profile retrieval action. Either a profile ID or a customer account value must be populated, but not both.	C	22	AN

Element Name	Description	Required	Max Char	Field Type
ccAccountNum	**Cardholder Account Number on Profile to be Retrieved** This value cannot be changed through a profile retrieval action. Either a profile ID or a customer account value must be populated, but not both.	C	19	AN

M = Mandatory, C = Conditional, O = Optional

Request sample – Fetch (Retrieve) Profile

```
{
    "bin": "000001",
    "merchantID": "041756",
    "customerRefNum": "1115504102"
}
```

Response elements – Create/Update/Fetch Profile

The table below lists the elements for a profile response.

Note: The response elements are the same for creating, updating and fetching profiles.

Field	Description	Required	Max Char	Field Type
version	**Version** Version of Simple Object Access Protocol (SOAP) Web Service Definition Language (WSDL) being used for the message. Latest version and recommended value: blank	O	5	AN
bin	**Transaction routing definition echoes the BIN sent in request.**	M	6	N
merchantID	**Gateway merchant account number assigned by Merchant Services**. Echoes the MID sent in request.	M	15	N
customerName	**Customer Billing Name** Echoes the customer name sent in the request.	O	30	AN

Field	Description	Required	Max Char	Field Type
customerRefNum	**Customer Reference Number** If this is the response to a profile add request and `customerProfileFromOrderInd` is **A**, this field returns a customer reference number assigned by the Orbital Gateway. Otherwise, this field echos the customer reference number sent in the profile request.	M	22	AN
profileAction	**Customer profile action that was requested**. • C = `customerProfileAdd` response • U = `customerProfileChange` response • R = `customerProfileFetch` response • D = `customerProfileDelete` response	M	1	A
procStatus	**Result Status of Profile Management** Communicates the success or failure of a profile management request. 0 = Success All other values constitute an error condition. Refer to the errors listed in Response Handling.	C	6	N
procStatusMessage	Text message associated with `procStatus` value.	M	Var	A

Field	Description	Required	Max Char	Field Type
customerAddress1	**Cardholder Billing Address Line 1** Data is conditionally returned if the request is `customerProfileFetch`, and data exists for the customer profile being retrieved.	C	30	AN
customerAddress2	**Cardholder Billing Address Line 2** Data is conditionally returned if the request is `customerProfileFetch`, and data exists for the customer profile being retrieved.	C	30	AN
customerCity	**Cardholder Billing City** Data is conditionally returned if the request is `customerProfileFetch`, and data exists for the customer profile being retrieved.	C	20	AN
customerState	**Cardholder Billing State** Data is conditionally returned if the request is `customerProfileFetch`, and data exists for the customer profile being retrieved.	C	2	AN

Field	Description	Required	Max Char	Field Type
customerZIP	**Cardholder Billing Address Zip Code** Data is conditionally returned if the request is `customerProfileFetch`, and data exists for the customer profile being retrieved.	C	10	AN
customerEmail	**Cardholder Email Address** Data is conditionally returned if the request is `customerProfileFetch`, and data exists for the customer profile being retrieved.	C	50	AN
customerPhone	**Cardholder Telephone Number** Data is conditionally returned if the request is `customerProfileFetch`, and data exists for the customer profile being retrieved.	C	14	N
customerCountryCode	**Cardholder Billing Country Code** Data is conditionally returned if the request is `customerProfileFetch`, and data exists for the customer profile being retrieved.	C	2	A

Field	Description	Required	Max Char	Field Type
profileOrderOverideInd	Dictates whether any order data can be pre-populated from the customer reference number (`customerRefNum`). Data is conditionally returned if the request is `customerProfileFetch`, and data exists for the customer profile being retrieved. • NO = No mapping to order data • OI = Use `customerRefNum` for `orderID` • OD = Use `customerRefNum` for comments • OA = Use `customerRefNum` for `orderID` and comments	M	2	A
orderDefaultDescription	**Order Description** Data is conditionally returned if the request is `customerProfileFetch`, and data exists for the customer profile being retrieved.	C	64	A
orderDefaultAmount	**Transaction Amount** Data is conditionally returned if the request is `customerProfileFetch` and the data exists for customer profile being retrieved.	C	12	N

Field	Description	Required	Max Char	Field Type
customerAccountType	**Customer's Payment Type** Data is conditionally returned if the request is `customerProfileFetch` and the data exists for customer profile being retrieved.	C	2	A
ccAccountNum	**Customer Credit Card Number** Data is conditionally returned if the request is `customerProfileFetch` and the data exists for customer profile being retrieved.	C	19	AN
ccExp	**Customer Credit Card Expiration Date** Data is conditionally returned if the request is `customerProfileFetch`, and the data exists for customer profile being retrieved.	C	6	N
mbType	**Managed Billing Type** • R = Recurring • D = Deferred	C	1	A
mbOrderIdGenerationMethod	**Managed Billing Order ID Generation Method** • IO = Customer reference number (profile ID) is used • DI = Dynamically generated order ID	C	2	A

Field	Description	Required	Max Char	Field Type
mbRecurringStartDate	**Managed Billing Recurring Start Date** Defines the date at which Orbital begins a recurring billing cycle with the associated profile. Format: MMDDYYYY	C	8	N
mbRecurringEndDate	**Managed Billing Recurring End Date** Defines the date at which the Orbital Gateway ends a recurring billing cycle with the associated profile. Format: MMDDYYYY	C	8	N
mbRecurringNoEndDateFlag	**Managed Billing "No End Date" Indicator** • Y = Recurring transactions are scheduled for an indefinite amount of time. A Y in this element overrides the value, if any, in the `mbRecurringEndDate` element. • N (or blank) = Orbital is using the value of the `mbRecurringEndDate` element to define the recurring end date.	C	1	A
mbRecurringMaxBillings	**Managed Billing Maximum Number of Billings** The maximum number of billings permitted for a recurring billing cycle. Valid values: 1–999999	C	6	N

Field	Description	Required	Max Char	Field Type
mbRecurringFrequency	**Managed Billing Recurring Frequency Pattern** This pattern is a subset of a standard CRON expression, comprising 3 fields separated by white space. For additional information regarding of these fields, as well as the usage of the special characters (with multiple example values), refer to the Profile Management and Managed Billing sections of this guide.	C	64	A
mbDeferredBillDate	**Managed Billing Deferred Billing Date** Format: MMDDYYYY	C	8	N
mbStatus	**Managed Billing Customer Status** Text message indicating the status of a managed billing request.	C	Var	A
softDescMercName	**Soft Descriptor Merchant Name**	C	25	A
softDescProdDesc	**Soft Descriptor Product Description**	C	18	A
softDescMercCity	**Soft Descriptor Merchant City**	C	13	A
softDescMercPhone	**Soft Descriptor Merchant Phone**	C	12	N
softDescMercURL	**Soft Descriptor Merchant URL**	C	13	AN

Field	Description	Required	Max Char	Field Type
softDescMercEmail	**Soft Descriptor Merchant Email**	C	13	AN
status	**Current Status of a Profile** • A = Active • I = Inactive • MS = Manual Suspend	C	Var	A
dpanInd	**CDPT Indicator** Used to identify the type of CDPT transaction submitted during an authorization. Valid value: S = For subsequent or recurring authorizations	O	1	A
mitMsgType	**CIT/MIT Code** Echoes the value in the request.	C	4	A
mitSubmittedTransactionID	**Submitted MIT TXID** Echoes the value in the request.	C	15	A
cardBrand	**Card Type/Brand for a Transaction**	C	2	A

Response sample – Create/Update/Fetch Profile


```
{
    "version": null,
    "bin": "000001",
    "merchantID": "041756",
    "customerName": "NEW CUSTOMER",
    "customerRefNum": "19143753",
    "profileAction": "READ",
    "procStatus": "0",
    "procStatusMessage": "Profile Request Processed",
    "customerAddress1": "123 ADDRESS1",
    "customerAddress2": "STE 350",
    "customerCity": "TAMPA",
    "customerState": "FL",
    "customerZIP": "33607",
    "customerEmail": "email@email.com",
    "customerPhone": "8001112222",
    "customerCountryCode": "US",
    "profileOrderOverideInd": "",
    "orderDefaultDescription": "test",
    "orderDefaultAmount": "1550",
    "customerAccountType": "CC",
    "ccAccountNum": "XXXXXXXXXXXX5454",
    "ccExp": "202109",
    "mbType": null,
    "mbOrderIdGenerationMethod": null,
    "mbRecurringStartDate": null,
    "mbRecurringEndDate": null,
    "mbRecurringNoEndDateFlag": null,
    "mbRecurringMaxBillings": null,
    "mbRecurringFrequency": null,
    "mbDeferredBillDate": null,
    "mbStatus": null,
    "softDescMercName": null,
    "softDescProdDesc": null,
    "softDescMercCity": null,
    "softDescMercPhone": null,
    "softDescMercURL": null,
    "softDescMercEmail": null,
    "status": null,
    "dpanInd": null,
    "mitMsgType": null,
    "mitSubmittedTransactionID": null,
    "cardBrand": null
}
```

Authorization, Profile Creation, and Managed Billing samples

Request sample – Google Pay


```
{
"audit" : {
    "latitudeLongitude": "1,1",
    "politicalTimeZone": "0500",
    "mobileDeviceType": 80
    },
"encryptedPaymentBundle": {
"signature":"MEQCID7me9PEtUNcra0pjwi5YLTx6J0AL/Yzc1s0aDIy85VQAiAmwHJexjH9J8UkvHS/SlfXIatAa3vk
Qq/kYWBFGN7Lcg\u003d\u003d","protocolVersion":"
ECv2","signedMessage":"{\"encryptedMessage\":\"TO8TPNwf7+gGlXvgg8i9b99b299kIpUVLOKRVFRIq4evzI
g8TbE9qY4gMKOHshy446STxo3FS1IAqA6hC9h/Q1EcT8nXrnYhymek0Cv1NcESC7r5Z7vvF10w9KPXOlYHZoiz2yEeJDE
m4f0F9v6XYUfw4J6GTvWZ/lYOXNv6j9D5855T+1ED7sXIZshzHofz9UbGLTb+/g2f8QpVzINQlW9dIQ9HmDsNXTT9ID2s
/SgdM7+wUyMRgSF746HuLZjQjVX7gV4Ag3EWqnl+FaJaMYKo5mDawGr0IVobWFiLHtEt7YVxvoV8+i9mdI9MESTFmiKZ9
9Vuy1eUoA6hsO8vHRnrnu3kyePuIvzQ7DkElprJ2EYiC17Ix+R4YzXDO9l1TQUHTKhozS2HLL17t/Nho+B0GWSgsLXZHE
PCaRmBkozt9D2gCvt03b5YRiYtsBmn0A\\u003d\\u003d\",\"ephemeralPublicKey\":\"BIrp+aB8f0Kpyymylyy
ns3+tHGwxlSiGnLP2unSWTcocg/EfCaFFbhOVMHHikI4Uv1kYCR66gX9KluAOQwA1/zM\\u003d\",\"tag\":\"/9QIu
+rwIEUr/9td5TE5u5pzEJ3HZwfZymOAJVm7fyY\\u003d\"}"
    },
"billAddress" : {
    "name": "Billing Address Name",
    "address1": "Billing Address 1",
    "address2": "Billing Address 2",
    "city": "Billing Address City",
    "state": "GA",
    "zip": "33711-4444",
    "countryCode": "US",
    "phone": "9998887777",
    "phoneType": "W"
    },
"managedBilling": {
    "mbType": "R",
    "mbOrderIdGenerationMethod": "IO",
    "mbRecurringStartDate": "01262020",
    "mbRecurringEndDate": "",
    "mbRecurringNoEndDateFlag": "Y",
    "mbRecurringMaxBillings": "",
    "mbRecurringFrequency": "? */5 MON",
    "mbDeferredBillDate": ""
},
"orderId": "123456",
"comments": "Comments",
"cardIndicators": "Y",
"partialAuthInd": "Y",
"bin": "000001",
"transactionAmount": 1000,
"walletType": "2"
"addProfileFromOrder": "A",
"profileOrderOverideInd": "NO",
"customerRefNum": "",
"mit" : {
     "mitMsgType": "CREC",
     "mitSubmittedTransactionID": "",
     "mitStoredCredentialInd": "Y"
},
"softDescriptor" : {
```

```
"merchantName": "Test Merchant",
"productDesc": "Pay1",
"merchantCity": "City",
"custServicePhone": "9998887777",
"merchantURL": "test.com",
"merchantEmail": "support@xyz.com"
}
```

Response sample – Google Pay

```
{
"ResponseCode": {
    "authorizationCode": "tst007",
    "visaVbVRespCode": "",
    "procStatus": "0",
    "procStatusMessage": null,
    "approvalStatus": "1",
    "respCode": "00",
    "respCodeMessage": "Approved",
    "hostResponseCode": "100",
    "hostResponseCodeMessage": null,
    "avsRespCode": "3 ",
    "hostAvsRespCode": "",
    "profileProcStatus": "0",
    "profileProcStatusMessage": "Profile Created"
},
"EnhancedAuth": {
    "ctiAffluentCard": "N",
    "ctiCommercialCard": "N",
    "ctiDurbinExemption": "Y",
    "ctiHealthcareCard": "N",
    "ctiLevel3Eligible": "N",
    "ctiPayrollCard": "X",
    "ctiPrepaidCard": "N",
    "ctiPINlessDebitCard": "N",
    "ctiSignatureDebitCard": "N",
    "ctiIssuingCountry": "USA"
},
"txRefNum": "5AD9DF5F7F91BB04241FC162CCD2A881C5E05365",
"remainingBalance": null,
"requestedAmount": 1000,
"redeemedAmount": 1000,
"partialAuthOccurred": null,
"hash1": "3adfe2a59aeede7b3d090c3b2293bb1e07c79e8015da69372c1d3d55078de29b",
"hash2": "a46014fle3a3018d44e5c5eecb80a7027cf8bce743eec713d7d8fe89fbf3778c",
"lastFourFPAN": "1111",
"lastFourDPAN": null,
"orderId": "123456",
"cardBrand": "VI"
"customerRefNum":"18888941",
"customerName":"",
"mitReceivedTransactionID": null
 }
```

Request sample – Apple Pay

```
{
"audit" : {
"latitudeLongitude" : "1,1",
"politicalTimeZone" : "0500",
"mobileDeviceType" : 80
},
"encryptedPaymentBundle" : {
```

```
  "data" :
"IzxSm6YWehmlLvk5HY/rsl4hhWuorOG7R6ERP0fqzTokMhS5JtyAU8ajPIu/aHcbOxYQOhvk/K+3n6N7SbEKgSuT100Y
FmeIKh3IkSLa4u1/1Y4Z9y5bqZFPxd8IcQnuR8HZKgJDHCXQzDDYP4JBMtqZQzRztzsIfa4eoOnGuZCc2s+WxGap4iv92
vPj8tAHonvSE9t0ByUCBLgfvu25GR0eJb6UM8nBvxP2/qBSElOuyLo80enrZ6tlp3xtpBEV8oeOc9iLSmalayfD7JQxZX
d2cWA/sZPWn4VGIj7Dt05NYE/iFZrw2VOa2hOJ4/4dOGS1KJzhw+RPRufhadAF96k7O3LwbMphcM9sZLN/Y/LSqVFGzIq
6ZlrnOwcxzvjNqw4ccNl4v3eehL4TRRgfF3LirV56BeADzJmq0pB3W/vu",
  "header" : {
    "ephemeralPublicKey" :
"MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEQ3CCwyRLUK61yxYifPLYY87iWcPydTCL0PpAOkpOAvZDCCffKbQTsxK9
707qmVrAmH0wDNZEbLJ9Ob3teiiCbA==",
    "publicKeyHash" : "MUwkjyUBpyRiZTVMUrIzA6+SIrr9mV8nNct6YO0rGNg="
  },
  "signature" :
"MIAGCSqGSIb3DQEHAqCAMIACAQExDzANBglghkgBZQMEAgEFADCABgkqhkiG9w0BBwEAAKCAMIIBYjCCAQigAwIBAgIG
AV1lOPsBMAoGCCqGSM49BAMDMDgxITAfBgNVBAMMGFBheW1lbnRlY2ggTW9iaWxlIFNESyBDQTETMBEGA1UECgwKUGF5b
WVudGVjaDAeFw0xNzA3MjExMjU2NTlaFw0zNDA3MjExMjU2NTlaMDgxITAfBgNVBAMMGFBheW1lbnRlY2ggTW9iaWxlIF
NESyBDQTETMBEGA1UECgwKUGF5bWVudGVjaDBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABEKuXMH9Q3bZlekeTuImojx
PuHQnxA4jIKiFwF3wOH6nQY94asOmLLLws3JD9tv2M2P7ppUl961r15aw48Gnr2UwCgYIKoZIzj0EAwMDSAAwRQIhAIvc
LMW83wgdvH0Mhi1ZJa93CV5bY6Ru5GKY/0vNb1F4AiBO4bPOqW7YR8GlJ6x823vx+AATTg5gocYGrj8tquPnjQAAMYIBG
zCCARcCAQEwQjA4MSEwHwYDVQQDDBhQYXltZW50ZWNoIE1vYmlsZSBTREsgQ0ExEzARBgNVBAoMClBheW1lbnRlY2gCBg
FdZTj7ATANBglghkgBZQMEAgEFAKBpMBgGCSqGSIb3DQEJAzELBgkqhkiG9w0BBwEwHAYJKoZIhvcNAQkFMQ8XDTE3MDk
xODEyMzA1NVowLwYJKoZIhvcNAQkEMSIEIFOTICKavR26ewV/9jepdbFWNoASpvLan5brcCutlZHzMAoGCCqGSM49BAMC
BEgwRgIhAO8S85/SS1fX0TRyDu7RA5wO/lRTF2ayk1PPcE9IN7i3AiEApAP4zETvW3jpipxp/nrKcISIGSm+XTmHXCiJZ
B/vthMAAAAAAAA=",
  "version" : "EC_v1"
},
"billAddress" : {
"name" : "Billling Address Name",
"address1" : "Billing Address 1",
"address2" : "Billing Address 2",
"city" : "Billing Address City",
"state" : "GA",
"zip" : "33711-4444",
"countryCode": "US",
"phone" : 9998887777,
"phoneType" : "W"
},
"managedBilling" : {
    "mbType": "R",
    "mbOrderIdGenerationMethod": "IO",
    "mbRecurringStartDate": "01262020",
    "mbRecurringEndDate": "",
    "mbRecurringNoEndDateFlag": "Y",
    "mbRecurringMaxBillings": "",
    "mbRecurringFrequency": "? */5 MON",
    "mbDeferredBillDate": ""
  },
"orderId": "Debundle1223",
"comments": "Comments",
"cardIndicators": "Y",
"recurringInd": "",
"partialAuthInd": "Y",
"walletType" : "1",
"addProfileFromOrder": "A",
"profileOrderOverideInd": "NO",
"customerRefNum": "",
"mit" : {
    "mitMsgType": "CINS",
    "mitSubmittedTransactionID": "",
    "mitStoredCredentialInd": "Y"
  }
}
```

Response sample – Apple Pay


```
{
"ResponseCode": {
    "authorizationCode": "tst949",
    "mcRecurringAdvCode": "",
    "visaVbVRespCode": "A",
    "procStatus": "0",
    "procStatusMessage": null,
    "approvalStatus": "1",
    "respCode": "00",
    "respCodeMessage": "Approved",
    "hostResponseCode": "100",
    "hostResponseCodeMessage": null,
    "avsRespCode": "B ",
    "hostAvsRespCode": "I3",
    "profileProcStatus": "0",
    "profileProcStatusMessage": "Profile Created"
},
"EnhancedAuth": {
    "ctiAffluentCard": "N",
    "ctiCommercialCard": "N",
    "ctiDurbinExemption": "N",
    "ctiHealthcareCard": "Y",
    "ctiLevel3Eligible": "N",
    "ctiPayrollCard": "X",
    "ctiPrepaidCard": "N",
    "ctiPINlessDebitCard": "N",
    "ctiSignatureDebitCard": "N",
    "ctiIssuingCountry": "USA"
},
"txRefNum": "5E2F4EC0142A11FD90377182ADFE0F7EBC47532F",
"remainingBalance": null,
"requestedAmount": 1000,
"redeemedAmount": 1000,
"partialAuthOccurred": null,
“hash1”: “3adfe2a59aeede7b3d090c3b2293bb1e07c79e8015da69372c1d3d55078de29b”,
“hash2”: “a46014fle3a3018d44e5c5eecb80a7027cf8bce743eec713d7d8fe89fbf3778c”,
"lastFourFPAN": null,
"lastFourDPAN": "9990",
"orderId": "Debundle1223",
"cardBrand": "VI"
"customerRefNum": "18888941",
"customerName": "",
"mitReceivedTransactionID": null
}
```

Response handling

ProcStatus codes

The **ProcStatus** element in the response provides an indication of whether or not the request was successful. A value of **0** indicates success, while any value greater than **0** indicates a failure. The table below provides a list of various **ProcStatus** codes that can be returned.

Code	Description and Occurrences
0	**Success**
8881	**Invalid Location** Occurs when a request is sent from restricted countries.
8882	**Missing or Incorrect BIN Value** Occurs if a merchant is sending the wrong `merchantId` or BIN value in the request body, or `merchantId` and BIN are not matching
8883	**Invalid User or Decryption Failed – Contact Merchant Services Support** Occurs when an encrypted payment bundle sent in the request contains an invalid user, Merchant Identifier (MID), or chain ID, or the request is not well formed.
8884	**System Error – Contact Merchant Services Support** Occurs when either a MID or other required elements in the request are missing.
8886	**Setup Error – Contact Merchant Services Support** Occurs when a merchant key is not properly configured in an Apple Pay payment bundle.
8887	**Merchant Not Enabled to Process [Wallet Type] Transactions** Occurs when a merchant is not enabled to process transactions for the wallet type sent in the request.
8889	**Incorrect Payment Action Indicator Value** Occurs when an incorrect value is sent in a payment action indicator.

Code	Description and Occurrences
8890	**Transaction Amount is Missing or Invalid** Occurs when a transaction amount is missing or invalid.
8892	**Mode in the payment container is not intended for this environment** Occurs if payment mode in encrypted payment bundle does not match the one expected in the environment (there are two modes, TEST and PROD).
8893	**Payload Missing Payment Context** Payload is missing `paymentContext` for wallet type. Occurs if `paymentContext` attribute is missing or empty in Merchant request body.
8894	Action Restricted for wallet type
20412	**Precondition Failed: Security Information is Missing.** Occurs when an invalid Orbital connection username or password is sent in the request.
20403	If a clear text request is made to one of the Orbital Gateway URLs, the Gateway returns an error condition (HTTP 403 error) with the accompanying eXtensible Markup Language (XML) payload containing a **ProcStatus 20403** error.
21010	An API key or call ID is not found, or data referenced by the API key or call ID is invalid or unavailable.
21015	**Invalid request data.** A required field is either missing or invalid.
21017	The API key used in the operation is not authorized for the requested action. Ensure the API key corresponds to the call ID.
21035	The shipping region is not accepted by the merchant.
21033	The requested data access level (i.e., data level) is invalid.
21058	The customer's account is locked.
21059	The customer's account is closed.
21073	Further operations on the card are not allowed.

Code	Description and Occurrences
21065	Expired call ID.
21076	The token-enabled card is not found and/or may be deleted.
21074	Invalid token request.

Error handling: profiles

The table below provides error handling codes for profiles.

Code	Description	Action
9549	Profile: Cannot %s profile for customer reference number: [%s] and MID: [%s]. Profile is not active.	Fix
9550	Profile: Invalid customer reference number from order indicator.	Fix
9551	Profile: Invalid customer reference number. The field is missing, invalid, or has exceeded the maximum length of [%s].	Fix
9552	Profile: System failure. Unable to perform the customer's profile request at this time.	Call
9553	Profile: Invalid action indicator: [%s]. Must be one of the following values: [%s].	Fix
9554	Profile: Invalid: [%s].	Call
9555	Profile: Invalid BIN: [%s]. The field is missing, invalid, or has exceeded the maximum length of: [%s].	Fix
9556	Profile: Invalid MID: [%s]. The field is missing, invalid, or has exceeded the maximum length of: [%s].	Fix
9557	Profile: Invalid name: [%s]. The field is missing, invalid, or has exceeded the maximum length of: [%s].	Fix
9558	Profile: Invalid address: [%s]. The field is missing, invalid, or has exceeded the maximum length of: [%s].	Fix

Code	Description	Action
9559	Profile: Invalid address 2: [%s]. The field is missing, invalid, or has exceeded the maximum length of: [%s].	Fix
9560	Profile: Invalid city: [%s]. The field is missing, invalid, or has exceeded the maximum length of: [%s].	Fix
9561	Profile: Invalid state: [%s]. The field is missing, invalid, or has exceeded the maximum length of: [%s].	Fix
9562	Profile: Invalid zip: [%s]. The field is missing, invalid, or has exceeded the maximum length of: [%s].	Fix
9563	Profile: Invalid email: [%s]. The field is missing, invalid, or has exceeded the maximum length of: [%s].	Fix
9564	Profile: Invalid phone: [%s]. The field is missing, invalid, or has exceeded the maximum length of: [%s].	Fix
9565	Profile: Invalid order description: [%s]. The field is missing, invalid, or has exceeded the maximum length of: [%s].	Fix
9566	Profile: Invalid amount: [%s]. The field is missing, invalid, or has exceeded the maximum length of: [%s].	Fix
9567	Profile: Invalid account type indicator: [%s]. Must be one of the following values: [%s]	Fix
9568	Profile: Invalid account number: [%s]. The field is missing, invalid, or has exceeded the maximum length of: [%s].	Fix
9569	Profile: Invalid account expire date: [%s]. The field is missing, not properly formatted, or has exceeded the maximum length of: [%s].	Fix
9570	Profile: Invalid ECP account DDA: [%s]. The field is missing or has exceeded the maximum length of: [%s].	Fix
9571	Profile: Invalid ECP account type indicator: [%s]. Must be one of the following values: [%s]	Fix

Code	Description	Action
9572	Profile: Invalid ECP account route: [%s]. The field is missing, invalid, or has exceeded the maximum length of: [%s].	Fix
9573	Profile: Invalid ECP bank payment delivery method: [%s]. Must be one of the following values: [%s]	Fix
9576	Profile: Unable to perform profile transaction. The associated transaction has failed.	Call
9577	Profile: Invalid order override indicator: [%s]. Must be one of the following values: [%s].	Fix
9578	Profile: Merchant BIN [%s]:[%s] is not allowed to perform profile transactions.	Call
9579	Profile: Merchant BIN [%s]:[%s] is not active.	Call
9580	Profile: Cannot %s profile for customer reference number: [%s] and MID: [%s]. A database error has occurred.	Call
9581	Profile: Cannot %s profile. Profile does not exist for customer reference number: [%s] and MID: [%s].	Fix
9582	Profile: Cannot %s profile. Profile already exists for customer reference number: [%s] and MID: [%s] %s.	Fix
9584	Profile: Missing electronic checking account information.	Fix
9585	Profile: Missing credit card account information.	Fix
9586	Profile: Profile request is detected, but has an action type of [I] (Ignore).	Call
9587	Profile: Auto-generation of customer reference number ERROR: Indicator: [%s]: customer reference number [%s] invalid when derived from [%s]. The field is missing, invalid, or has exceeded maximum length of [%s].	Call
9588	Profile: Unable to determine "on-the-fly" profile action: Customer reference from profile Ind: [%s] and customer reference number is [%s]. One of the values must be valid.	Fix
9589	Profile: cannot %s profile: A customer profile name is required.	Fix

Code	Description	Action
9590	Profile: cannot %s profile: A customer zip code is required.	Fix
9591	Profile: Profile merging mismatch error: stored profile account type of: [%s] does not match incoming account type of: [%s]	Fix
9592	Invalid profile status requested.	Fix
9593	Profile: Invalid country code [%s]. Supported values are **[CA]**, **[GB]**, **[UK]**, or **[US]**.	Fix
9594	Profile status is currently [%s]. Only refunds are available while the profile is in this status.	Fix

Error handling: Managed Billing

The table below provides error handling codes for managed billing.

Code	Description	Action
9850	Managed billing features are not supported for PIN-less debit transaction types.	Fix
9851	Merchant’s account is not configured to use managed billing features.	Call
9852	Profile level for merchant account is set to **chain** level. In order to use managed billing, the profile level must be set to **merchant** level.	Call
9853	Invalid order ID generation method. Use a valid value.	Fix
9854	Invalid managed billing type for merchant.	Call
9855	Managed billing: Dollar value for a micro payment cannot be less than $1 - [%s].	Fix
9856	Managed billing: Micro payment transaction dollar amount for Tandem must be numeric, and has maximum value of [99999999] - [%s].	Fix
9857	Managed billing: Micro payment transaction dollar amount for Stratus must be numeric, and has maximum value of [999999999999] - [%s].	Fix

Code	Description	Action
9858	Managed billing: Maximum micro payment billing days is [1 - 30] - [%s].	Fix
9859	Managed billing: Maximum micro payment transactions before acquiring funds is [1 - 99] - [%s].	Fix
9860	Managed billing: Micro payment profiles require at least one trigger method [maximum dollar amount, maximum billing days, maximum transactions].	Fix
9861	Deferred billing date must be a valid date (at least 1 day, and at most 365 days, in the future).	Fix
9862	Recurring start date must be a valid date at least 1 day in the future.	Fix
9863	Only one recurring end date trigger can be selected.	Fix
9864	Invalid recurring no end date flag. Must be **Y** or **N**.	Fix
9865	Invalid maximum number of recurring billings.	Fix
9866	Recurring end date must be a valid date at least 1 day greater than the recurring start date.	Fix
9867	One of 3 available recurring triggers must be set.	Fix
9868	Invalid recurring format.	Fix
9869	Industry type of **IN** can only be used when a merchant is configured for a managed billing type of recurring.	Fix
9870	The order ID generation method must be Dynamic [DI] for micro payments.	Fix
9871	Missing default managed billing values. All values must be set in transaction payload.	Fix
9872	The Industry Type [IN] is only valid for managed billing transactions.	Fix
9873	The cancel date must be valid.	Fix
9874	Daily frequency patterns are not accepted.	Fix
9875	Scheduling is not complete. Contact Merchant Services.	Call

Code	Description	Action
9876	The profile is locked for an in-progress update.	Call
9877	Send in future payment cancel requests **after** updating profile.	Fix
9878	A future payment date could not be found to cancel.	Fix
9879	A cancelled payment date could not be found to restore.	Fix
9880	The start and end date ranges are too small for the selected recurring frequency (i.e., there are no possible future billings).	Fix
9881	An existing deferred payment is already in progress.	Fix
9882	The user does not have the privileges necessary to set-up a managed billing profile.	Call
9883	The industry type of recurring is not allowed to be set-up as deferred managed billing type.	Fix
9884	An error occurred while searching for the transaction related to the retry trace ID.	Call
9885	Failed to find a transaction associated with retry trace ID.	Fix
9886	Managed billing: PIN-less transactions can only be used with a recurring billing type.	Call
9887	Inquiry: Transaction using retry trace number [%s] is in progress.	Fix
9888	Invalid value for card indicators: [%s]. Valid values are **Y**, **N** or **EMPTY**.	Fix
9889	Inquiry: The original transaction resulted in an error.	Fix

Error handling: MIT

The table below provides error handling codes for MIT.

Code	Description	Action
19811	MIT profile cannot be used in non-MIT scenario	Fix
19812	MIT profile was created for [**MIT CODE STORED IN PROFILE**]. Only the following values are allowed in an MIT message type: [MUSE, CUSE, MRAU, MRSB, MREC, and MINS] (for VI). MIT profile was created for [**MIT CODE STORED IN PROFILE**]. Only the following values are allowed in an MIT message type: [MRAU, MREC] (for Discover [DI]).	Fix
19794	MIT: Invalid stored credential flag [**incorrect value submitted in request**].	Fix
19796	MIT: Original Transaction Identifier (TXID) is required for merchant-initiated transactions.	Fix
19797	MIT: Invalid original TXID [**123456789012*4a**], special characters are not permitted.	Fix
19798	MIT: Original TXID is not expected for customer-initiated transactions.	Fix
19793	MIT: Invalid MIT message type [MIT code passed in request] [Only CSTO, CGEN, CREC, CINS, MUSE, MREC, MINS are allowed while using non-MIT Profile] (for VI). Invalid MIT message type [MIT code passed in request] [Only CGEN, CREC and MREC are allowed while using NON MIT Profile] (for DI).	Error
19810	MIT: TXID is required to create MIT profile.	Error
19813	MIT profile contains invalid MIT message type [**MIT CODE STORED IN PROFILE**]. Only [CSTO, CGEN, CREC and CINS] are allowed (for VI). MIT profile contains invalid MIT message type [**MIT CODE STORED IN PROFILE**]. Only [CGEN, CREC] are allowed (for DI).	Error

Code	Description	Action
19814	New MIT profile cannot be added while using an existing MIT profile [**Profile ID sent in the request**].	Error

Response code values

The table below provides the response code values.

respCode	Definition	Status	Action	Host Code Stratus	Host Code Tandem
00	Approved	Approved	None	100, 102	00, 100, 102
01	Call/Refer to card issuer	Decline	Voice	401	01
02	Refer to card issuer's special conditions	Decline	Voice	N/A	02
03	Invalid merchant number	Error	Fix	231	03
04	Pickup	Decline	Cust.	501	04
05	Do not honor	Decline	Cust.	530	05
06	Other error	Decline	Cust.	594	06
07	Stop deposit order	Decline	Cust.	570	N/A
08	Approved authorization, honor with identification	Approved	None	N/A	08
09	Revocation of authorization	Decline	Cust.	571	N/A
10	Default call	Decline	Voice	402	N/A
11	Approved authorization, VIP approval	Approved	None	N/A	11
12	Invalid transaction type	Decline	Cust.	606	12

respCode	Definition	Status	Action	Host Code Stratus	Host Code Tandem
13	Bad amount	Decline	Fix	592	13
14	Invalid credit card number	Decline	Fix	591	14
15	Default call low fraud	Decline	Voice	442	N/A
16	Default call medium fraud	Decline	Voice	443	N/A
17	Default call high fraud	Decline	Voice	444	N/A
18	Default call unavailable fraud	Decline	Voice	445	N/A
19	Re-enter transaction	Error	Resend	N/A	19
20	Floor low fraud	Decline	Cust.	332	N/A
21	Floor medium fraud	Decline	Cust.	333	N/A
22	Floor high fraud	Decline	Cust.	334	N/A
23	Floor unavailable fraud	Decline	Cust.	335	N/A
24	Validated	Approved	None	101	101
26	Pre-noted	Approved	None	103	103
27	No reason to decline	Approved	None	104	N/A
28	Received and stored	Approved	None	105	N/A
29	Provided authorization	Approved	None	106	N/A
30	Invalid value in message	Error	Fix	225	30
31	Request received	Approved	None	107	N/A

respCode	Definition	Status	Action	Host Code Stratus	Host Code Tandem
32	BIN alert	Approved	None	110	N/A
33	Card is expired	Decline	Cust.	522	33
34	Approved for partial	Approved	None	111	N/A
35	Zero amount	Error	Fix	203	N/A
36	Bad total authorization amount	Error	Fix	205	N/A
37	Invalid secure payment data	Error	Fix	245	N/A
38	Merchant not MC **SecureCode** enabled	Decline	Call	246	N/A
39	Previously processed transaction	Error	Fix	109	N/A
40	Requested function not supported	Error	Call or Fix	N/A	40
41	Lost/Stolen	Decline	Cust.	502	N/A
42	Account not active	Decline	Cust.	N/A	15
43	Lost/Stolen card	Decline	Cust.	N/A	43
44	Account not active	Decline	Cust.	N/A	N/A
45	Duplicate transaction	Decline	Cust.	551	N/A
46	Blanks not passed in **Reserved** field	Decline	Fix	248	N/A
50	Positive ID	Decline	Cust.	802	N/A
52	Processor decline	Decline	Cust.	303	N/A
56	Restraint	Decline	Cust.	806	N/A

respCode	Definition	Status	Action	Host Code Stratus	Host Code Tandem
58	Transaction not permitted to terminal	Error	Call	N/A	58
59	Soft Address Verification Services (AVS)	Decline	Cust.	260	N/A
60	Additional customer authentication is required	Decline	Cust.	532	N/A
61	Do not honor, medium fraud	Decline	Cust.	533	N/A
62	Do not honor, high fraud	Decline	Cust.	534	N/A
63	Do not honor, unavailable fraud	Decline	Cust.	535	N/A
64	Card Verification Value (CVV)2/Card Verification Code (CVC)2 failure	Decline	Cust.	531	N/A
65	Invalid AX Card Identifier (CID)	Decline	Cust.	811	N/A
66	Other error	Error	Fix	204	N/A
68	Invalid CC number	Error	Fix	201	N/A
69	Does not match Method of Payment (MOP)	Error	Fix	233	N/A
71	No account	Decline	Fix	825	N/A
72	Invalid institution code	Decline	Fix	602	N/A
73	Method of payment is invalid for merchant	Error	Fix	834	834
74	Invalid expiration date	Decline	Cust.	605	54
75	Bad amount	Error	Fix	202	N/A

respCode	Definition	Status	Action	Host Code Stratus	Host Code Tandem
77	Invalid amount	Decline	Fix	607	N/A
78	Missing companion data	Error	Fix	227	N/A
79	Invalid merchant	Error	Fix	833	N/A
80	Invalid Method of Payment (MOP) for division	Error	Fix	239	N/A
81	Call low fraud	Decline	Voice	432	N/A
82	Call medium fraud	Decline	Voice	433	N/A
83	Call high fraud	Decline	Voice	434	N/A
84	Call unavailable fraud	Decline	Voice	435	N/A
85	Duplicated order number	Error	Fix	234	N/A
86	Authorization recycle host down	Error	Wait	236	N/A
87	Invalid currency	Error	Fix	238	N/A
89	Credit floor	Decline	Cust.	302	N/A
91	Approved low fraud	Approved	None	112	N/A
92	Approved medium fraud	Approved	None	113	N/A
93	Approved high fraud	Approved	None	114	N/A
94	Approved fraud service unavailable	Approved	None	115	N/A
95	Invalid data type	Error	Fix	226	N/A
96	Invalid record sequence	Error	Fix	228	N/A

respCode	Definition	Status	Action	Host Code Stratus	Host Code Tandem
97	Percentage does not total 100	Error	Fix	229	N/A
98	Issuer unavailable	Decline	Resend	301	N/A
99	No answer/unable to send	Error	Resend	000	99
A1	Payments not total order	Error	Fix	230	N/A
A2	Bad order number	Error	Fix	232	N/A
A3	FPO locked	Error	Wait	235	N/A
A4	FPO not allowed	Error	Call	237	N/A
A5	Authorization amount wrong	Error	Fix	240	N/A
A6	Illegal action	Error	Fix	241	N/A
A8	Invalid start date	Error	Fix	251	N/A
A9	Invalid issue number	Error	Fix	252	N/A
B1	Invalid transaction type	Error	Fix	253	N/A
B2	Account previously activated	Decline	Cust	580	16
B3	Unable to void transaction	Error	Fix	581	18
B5	Not on file	Decline	Fix	304	N/A
B7	Fraud	Decline	Cust.	503	N/A
B8	Bad debt	Decline	Cust.	504	N/A
B9	On negative file	Decline	Cust.	505	N/A

respCode	Definition	Status	Action	Host Code Stratus	Host Code Tandem
BA	Under 18 years old	Decline	Cust.	540	N/A
BB	Possible compromise	Decline	Cust.	541	N/A
BC	Bill to not equal to ship to	Decline	Cust.	542	N/A
BD	Invalid pre-approval number	Decline	Cust.	543	N/A
BE	Invalid email address	Decline	Cust.	544	N/A
BF	PA ITA number inactive	Decline	Cust.	545	N/A
BG	Blocked account	Decline	Cust.	546	N/A
BH	Address verification failed	Decline	Cust.	547	N/A
BI	Not on credit bureau	Decline	Cust.	548	N/A
BJ	Previously declined	Decline	Cust.	549	N/A
BK	Closed account, new account closed	Decline	Cust.	550	N/A
BL	Re-authorization	Decline	Cust.	560	N/A
BM	Re-authorization – No match	Decline	Cust.	561	N/A
BN	Re-authorization – timeframes exceeded	Decline	Cust.	563	N/A
BO	Stand In rules	Decline	Cust.	905	N/A
BP	Customer service phone number required on transaction types 1 (MO/TO) and 2 (recurring) MC only	Error	Fix	257	N/A

respCode	Definition	Status	Action	Host Code Stratus	Host Code Tandem
BQ	Issuer has flagged account as suspected fraud (DI only)	Decline	Cust.	596	N/A
BR	Invalid Merchant Category Code (MCC) sent	Error	Fix	249	N/A
BS	New card issued	Decline	Cust.	595	N/A
BT	Not authorized to send record	Decline	Fix	258	N/A
C1	Invalid issuer	Decline	Cust.	506	N/A
C2	Invalid response code	Decline	Fix	507	N/A
C3	Excessive PIN try	Decline	Cust.	508	N/A
C4	Over limit	Decline	Cust.	509	N/A
C5	Over freq limit Note: Additional customer authentication required for MC only.	Decline	Cust.	510	N/A
C6	Over Sav Limit	Decline	Cust.	511	N/A
C7	Over Sav freq	Decline	Cust.	512	N/A
C9	Over credit freq	Decline	Cust.	514	N/A
D1	Invalid for credit	Decline	Fix	515	N/A
D2	Invalid for debit	Decline	Fix	516	N/A
D3	Rev exceed withdrawal	Decline	Cust.	517	N/A
D4	One purchasing limit	Decline	Cust.	518	N/A

respCode	Definition	Status	Action	Host Code Stratus	Host Code Tandem
D5	On negative file	Decline	Cust.	519	519
D6	Changed field	Decline	Fix	520	N/A
D7	Insufficient funds	Decline	Cust.	521	N/A
D8	Encrypted data bad	Decline	Fix	523	96
D9	Altered data	Decline	Fix	524	N/A
E3	Invalid prefix	Decline	Fix	601	N/A
E4	Invalid institution	Decline	Fix	603	N/A
E5	Invalid cardholder	Decline	Fix	604	N/A
E6	BIN block	Decline	Fix	610	N/A
E7	Stored	Approved	None	704	N/A
E8	Invalid transit routing number	Error	Fix	750	750
E9	Unknown transit routing number	Error	Fix	751	751
F1	Missing name	Error	Fix	752	N/A
F2	Invalid account type	Error	Fix	753	N/A
F3	Account closed	Error	Cust.	754	754
F4	No account/unable to locate	Error	Fix	755	755
F5	Account holder deceased	Error	Cust.	756	756
F6	Beneficiary deceased	Error	Cust.	757	757

respCode	Definition	Status	Action	Host Code Stratus	Host Code Tandem
F7	Account frozen	Error	Cust.	758	758
F8	Customer opt out	Error	Cust.	759	759
F9	Automated Clearing House (ACH) non-participant	Error	Cust.	760	760
G1	No pre-note	Error	Fix	761	N/A
G2	No address	Error	Fix	762	N/A
G3	Invalid account number	Error	Fix	763	763
G4	Authorization revoked by consumer	Error	Cust.	764	764
G5	Customer advises not authorized	Error	Cust.	765	765
G6	Invalid Country/County Extended Code Page (CECP) action code	Error	Fix	766	N/A
G7	Invalid account format	Error	Fix	767	767
G8	Bad account number data	Error	Fix	768	N/A
G9	No capture	Decline	N/A	801	N/A
GA	Account non-convertible	Decline	N/A	769	769
H1	No credit function	Decline	N/A	803	N/A
H2	No debit function	Decline	N/A	804	N/A
H3	Rev exceed withdrawal	Decline	Cust.	805	N/A
H4	Changed field	Decline	N/A	807	N/A

respCode	Definition	Status	Action	Host Code Stratus	Host Code Tandem
H5	Terminal not owned	Decline	N/A	808	N/A
H6	Invalid time	Decline	Fix	809	N/A
H7	Invalid date	Decline	Fix	810	N/A
H8	Invalid terminal number	Decline	Fix	812	N/A
H9	Invalid PIN	Decline	Cust.	813	38
I1	Block activation failed – card range not set up for MOD 10.	Error	Fix	582	N/A
I2	Block activation failed – email or fulfillment flags were set to **Y**.	Error	Fix	583	N/A
I3	Declined – issuance does not meet minimum amount	Declined	Cust	584	N/A
I4	Declined – no original authorization found	Decline	Cust	585	N/A
I5	Declined – outstanding authorization, funds on hold	Decline	Cust	586	N/A
I6	Activation amount incorrect	Decline	Fix	587	N/A
I7	Block activation failed – account not correct or block size not correct.	Decline	Fix	588	N/A
I8	Mag stripe Card Verification Data (CVD) value failed	Decline	Fix	589	N/A
I9	Maximum redemption limit met	Decline	Fix	590	N/A
J1	No manual key	Decline	Fix	814	N/A

respCode	Definition	Status	Action	Host Code Stratus	Host Code Tandem
J2	Not signed in	Decline	Fix	815	N/A
J3	Excessive PIN tries	Decline	Cust.	816	N/A
J4	No DDA	Decline	Fix	817	N/A
J5	No SAV	Decline	Fix	818	N/A
J6	Excess DDA	Decline	Cust.	819	N/A
J7	Excess DDA FREQ	Decline	Cust.	820	N/A
J8	Excess SAV	Decline	Cust.	821	N/A
J9	Excess SAV FREQ	Decline	Cust.	822	N/A
K1	Excess card	Decline	Cust.	823	N/A
K2	Excess card freq	Decline	Cust.	824	N/A
K3	Reserved future	Decline	N/A	826	N/A
K4	Reserved closing	Decline	N/A	827	N/A
K5	Dormant	Decline	Cust.	828	N/A
K6	Non-Sufficient Funds (NSF)	Decline	Cust.	829	N/A
K7	Future Receiving Depository (RD) six	Decline	N/A	830	N/A
K8	Future RD seven	Decline	N/A	831	N/A
K9	Transaction code conflict	Decline	Fix	832	N/A
L1	In progress	Decline	Wait	901	N/A

respCode	Definition	Status	Action	Host Code Stratus	Host Code Tandem
L2	Process unavailable	Error	Resend	902	N/A
L3	Invalid expiration	Error	Fix	903	N/A
L4	Invalid effective	Error	Fix	904	N/A
L5	Invalid issuer	Decline	Fix	N/A	15
L6	Transaction not allowed for cardholder	Decline	Cust.	N/A	57
L7	Unable to determine network routing	Error	Call	N/A	92
L8	System error	Error	Call	N/A	97
L9	Database error	Error	Call	N/A	98
M1	Merchant override decline	Decline	Cust.	Merchant Selectable Response	Merchant Selectable Response
M2	Partial authorization not allowed	Decline	Cust	Partial Authorization Support	Partial Authorization Support
ND	Account number appears on European direct debit negative file	Decline	Cust	719	N/A
P1	Electronic Check Processing (ECP) – advanced verification service – account status verification and/or Account Owner Authorization (AOA) data is in a positive status.	Approved	None	116	N/A
P2	ECP account verification/AOA decline	Decline	Cust.	575	N/A

respCode	Definition	Status	Action	Host Code Stratus	Host Code Tandem
P3	No information found	Decline	Cust.	576	N/A
P4	ECP account verification decline	Decline	Cust.	578	N/A
P5	Not ACH eligible	Decline	Cust.	579	N/A
PA	Partial approval	Approved	N/A	N/A	10
PB	Revocation of all authorization	Decline	Cust.	572	17
PC	Country on fraud filter list	Decline	Cust	271	N/A
PD	Partial authorization override not allowed	Decline	Cust.	263	N/A
PP	No match for debit authorization based on trace, account, and division number	Error	Fix	N/A	N/A
PQ	Unable to validate debit Authorization record based on amount, action code, and MOP	Error	Fix	N/A	N/A
PR	Refund not allowed – refund requested on a star only BIN, or BIN not found	Error	Fix	599	N/A
R1	Blocked card number prefix	Decline	Cust.	269	N/A
R2	Blocked card number	Decline	Cust.	270	N/A
R3	Blocked issuing country	Decline	Cust.	271	N/A
R4	Ceiling limit	Decline	Cust.	275	N/A
R5	Not authorized to send record	Decline	Cust	258	N/A

respCode	Definition	Status	Action	Host Code Stratus	Host Code Tandem
R6	Authorization not found	Decline	Cust.	307	N/A
R7	Amount mismatch	Decline	Cust.	306	N/A
R8	Already reversed or nothing to reverse	Decline	Cust.	305	N/A
R9	Authorization code or response date invalid	Decline	Cust.	262	N/A
S1	Electronic processing not supported	Decline	Cust.	747	N/A
Z6	Additional authentication needed Note: For VI only.	Decline	Cust.	N/A	36

Additional references

Refer to the specifications available in Developer Center for details that describe host transaction processing.

Card Not Present (CNP) Stratus References

- Online Processing Developer Guide
- 120-Byte Batch Processing Developer Guide

Retail Host Tandem References

- PNS ISO Terminal Capture and Host Capture Developer Guides
- TCS Batch File Developer Guide
- UTF Host Capture Developer Guide

Developer Links from Wallet Providers

- Apple Pay
    - https://developer.apple.com/apple-pay/
    - https://developer.apple.com/support/apple-pay-sandbox/
    - Get support with Apple Pay implementation: https://getsupport.apple.com/?caller=wwdr&PGF=PGF90001
- Google Pay
    - https://developers.google.com/pay/api/
    - Get support with Google Pay implementation: https://developers.google.com/pay/api/support

Appendix A: details of encrypted payload

Apple Pay

The table below provides the encrypted payload details for Apple Pay.

Key	Type	Description
applicationPrimaryAccountNumber	string	Device-specific account number of the card used to fund the transaction.
applicationExpirationDate	date as a string	Card expiration date in YYMMDD format.
currencyCode	string	International Organization for Standardization (ISO) 4217 numeric currency code (as a string to preserve leading zeros).
transactionAmount	number	Transaction amount
cardholderName	string	Cardholder name. Optional: Must be requested within Apple Pay payment request class.
deviceManufacturerIdentifier	string	Hex-encoded device manufacturer identifier.
paymentDataType	string	Either 3-D Secure, or if using Apple Pay in China, Europay, Mastercard and Visa (EMV).
paymentData	payment data dictionary	Detailed payment data.

Google Pay

Google sends Digital Primary Account Number (DPANs) and Funding Primary Account Number (FPANs) in Google Pay payloads, depending on the device used by the customer to make the transaction. The merchant must make an election to receive DPANs and FPANs, or DPANs only when completing the setup with Google. The contents of the payload on browser-initiated transactions are noted in the **PAN Only** table. The contents of the payload on a device-initiated transaction are noted in the **Cryptopgram_3D** table.

PAN_ONLY

PAN_ONLY data is returned when a payment request originates from a web-based Google Pay interface. The table below provides the encrypted payload details for Google Pay **PAN_ONLY**.

Key	Type	Description
pan	string	Personal Account Number (PAN) is charged. This string contains digits only.
expirationMonth	number	The expiration month of the card, where 1 is January, 2 is February, etc.
expirationYear	number	The four-digit expiration year of the card (e.g., 2020).
authMethod	string	The authentication method of the card transaction.

CRYPTOGRAM_3DS

CRYTOGRAM_3DS data is returned when a payment request originates from a secured device (e.g., mobile, watch, tablet) using the Google Pay interface. The table below provides the encrypted payload details for Google Pay **CRYPTOGRAM_3DS**.

Key	Type	Description
cryptogram	string	A 3-D Secure cryptogram
eciIndicator	string	Not always present. Returned only for tokens on the Visa (VI) card network. This value is sent in the payment authorization request.
pan	string	PAN is charged. This string contains only digits.
expirationMonth	number	The expiration month of the card, where 1 is January, 2 is February, etc.
expirationYear	number	The four-digit expiration year of the card (e.g., 2020).
authMethod	string	The authentication method of the card transaction.